תרגילי מעבדה בחומרת
מיקרו-מחשב

אורט ישראל

תרגילי מעבדה בחומרת
מיקרו-מחשב

אריה נחום

לתירגול על גבי הערכה
לחומרת מיקרו מחשב 6800 ו-8085

תוצרת המרכז הפדגוגי



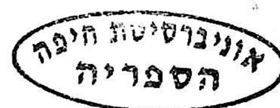
תכנון ופיתוח ערכת התרגילים : אריה נחום

כתיבת הניסויים : אריה נחום

סרטוט ועריכה גרפית : אביבה בן-דוד

עריכה : אביבה בן-דוד

הדפסה : אורית נחום





30/43

תוכן העניינים

עמוד

חוברת ניסויים זו משלימה את ערכת התרגול בחומרה של מיקרו-מחשבים, שפותחה במרכז הפדגוגי. יחד הן מהוות מערכת המאפשרת הוספת שלב בלימוד המיקרו-מחשב - לימוד ותרגול חומרה של מיקרו-מחשבים. היא מאפשרת להכיר את הבעיות השונות הכרוכות בבניית החומרה, איתור תקלות והפתרונות השונים. אין אפשרות, כמובן, ללמד את כל אופני הבניה והפתרונות, אבל המטרה היא לתת את העקרונות שלהם.

שלב לימוד זה, בא לאחר שלב לימוד תוכנה של מיקרופרוססורים וכתיבת תוכניות בשפת מכונה. אפיון פתרון לבעיה במיקרו-מחשב כרוך בשלוב של חומרה ותוכנה. גם איתור תקלות במערכת מיקרו-מחשב, כרוך בכתיבת תוכנית מתאימה. זו הסבה שאין אפשרות לתרגל חומרה ללא לימוד קודם של תוכנה למיקרו פרוססורים.

מערכת התרגול בחומרה של מיקרו-מחשבים באה להשלים מערכות מיקרו-מחשבים הקיימות בבתי הספר ולהתחבר אליהם. כתיבת התוכניות תתבצע על המערכות השונות ותרגול החומרה יתבצע על ערכת תרגול החומרה.

מה שהנחה אותנו באיפיון המערכת היו הנקודות הבאות:

1. בבתי הספר יש מערכות מחברות שונות. רובן מבוססות על ה-8085 ועל ה-6800. מערכת התרגול צריכה לאפשר התחברות למערכות הקיימות בבתי הספר.
2. שני המיקרופרוססורים האלה מיצגים את שתי הגישות הקיימות בתכנון מיקרופרוססורים ובעיקר בתכנון חומרה למיקרו-מחשבים. כמערכת חינוכית עלינו להראות את שתי הגישות ולא ללמד אחד משני המיקרופרוססורים. גם אם בבית הספר מלמדים מיקרופרוססור אחד, על המורה והתלמיד להיות מודעים להבדלי הגישה.
3. חיבור רכיבי החומרה יעשה על לוח תקיעה (מטריצה). בנוסף לכך תספק המערכת רכיבים כמו תצוגת 7-SEG., לדים ומפסקים. המערכת תספק גם נקודות חיבור ל-BUS המיקרופרוססור ומספר מעגלי עזר שונים כמו מעגל פענוח לרכיבי I/O, פורטי מבוא, פורטי מוצא ו-2 מתנדים (אחד קבוע ואחד משתנה).

נקודות איפיון אלה חייבו פתוח מערכת גמישה. התאמת המערכת למערכת מיקרו-מחשב מסויימת, נעשית על ידי בחירת כרטיס תיאום מתאים וקביעת גשרים במערכת באופן חד פעמי.

במקרה של חיבור מערכת מיקרו-מחשב עם 8085, BUS המיקרופרוססור שיתקבל במערכת יהיה ה-BUS לאחר ביצוע נעילה (LATCHING) של קוי הכתובת הנמוכים והפרדתם מקוי הנתונים. כלומר, נקבל 16 קוי כתובת נפרדים ל-8 קוי נתונים. ההפרדה מתבצעת על כרטיס התיאום במקרה ולא נעשתה במערכת המיקרו-מחשב.במידה וקוי ה-BUS יוצאים ממערכת המיקרו-מחשב ללא חוצצים, יהיו חוצצים על כרטיס התיאום.

המערכת, מרכיביה ואופן השימוש בהם, מתוארים בשלבים בחלקים העיוניים של הניסויים בהתאם לשלבי השמוש בהם.

חוברת הניסויים כוללת שמונה פרקים ובהם עשרים ושמונה נסויים. בכל פרק חלק עיוני וחלק של תרגול מעשי. החומר העיוני אינו בא במקום שעורים עיוניים, והוא רלוונטי לנסויים עצמם. החומר העיוני נלקח ברובו מספרו של המחבר "יסודות המיקרו-מחשב - חומרה ויישומים".

ביצוע הניסויים בכל פרק יקח מספר שעורי מעבדה. בכל ניסוי קיים חלק של כתיבת תוכנה. בחלקים העיוניים מוצעות תוכניות. התוכניות מופיעות כתרשימי זרימה או כתובות גם באסמבלי. על התלמיד להמיר תרשימים אלה לתוכניות, ולכתוב כתובות וצפנים בהתאם לסוג מערכת המיקרו-מחשב שברשותו. רצוי לבצע תרגילים בתוכנה מעבר לניתן בחוברת. יש להקפיד על רישום מפורט ומסודר של כתיבת התוכניות, כדי להקטין למינימום את שלב איתור התקלות בניסוי עצמו.

ברוב הניסויים התרגיל הוא בניית מערכת כלשהי, ובדרך כלל ללא תוצאות לעבוד. חשוב שהתלמיד יכין דו"ח מפורט הכולל תיאור הניסוי, מעגל חשמלי מלא, תוכניות שרשם בצורה מלאה כולל הערות ותיאור תוצאות הניסוי.

הכרת איפיון רכיב ואופן השימוש בו צריך להתבצע על ידי קריאת דפי נתונים. חשיבות רבה יש להעניק לנושא זה, עקב המיגוון העצום של רכיבים הקיימים במיקרו-מחשבים וצורות השימוש השונות בהם.

נשמח לקבל הערות והצעות לשפור.

אריה נחום.

פרק 1 - דיאגרמת זמנים 6800 ו-8085

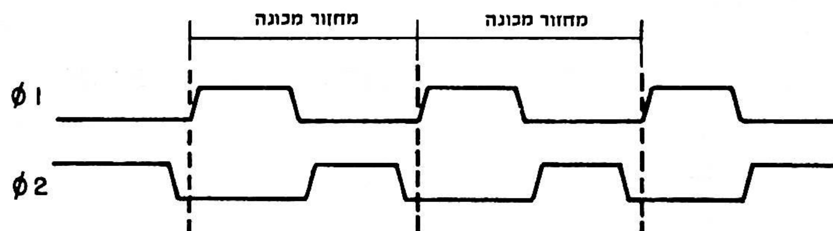
1.1 נושאי הפרק

א. דיאגרמות זמנים של ה-6800
ב. דיאגרמות זמנים של ה-8085
ג. אופן בדיקת דיאגרמת זמנים של מחזור מכונה מסוים.

רקע עיוני

1.2 דיאגרמת זמנים של ה-6800

ה-CPU הוא מערכת סינכרונית, המשנה את מצבה בזמני מעבר של דפקי שעון. ל-6800 2 כניסות לדפקי שעון. אותות השעון הם גלים רבועיים הפוכים במופע ולא חופפים (NON OVERLAPPING) כמתואר בציור.



ציור 1 - 1 : אותות השעון של ה-6800

צורה זו של דפקי שעון יוצרת ארבעה מצבי מעבר שונים במשך מחזור מכונה אחד. במחזור מכונה מבצע ה-CPU יחידת פעולה אחת. ב-6800 מחזור שעון הוא גם מחזור מכונה (דבר המאפשר לחשב בקלות זמן בצוע תוכנית). רכיבי התמיכה של ה-6800, במערכת המיקרו-מחשב, מסתנכרנים אל שעון המערכת (מתאימים את תזמון פעולותיהם אליו).

הפעולות הפנימיות של ה-CPU במחזור מכונה, מתחילות בעלית 01 ומסתיימות בירידת 02.

מבחינתה של מערכת חיצונית ל-6800 או רכיב תמיכה כלשהו (RAM, ROM, PORT וכד'), יש ל-CPU שלשה סוגים ש מחזורי מכונה.

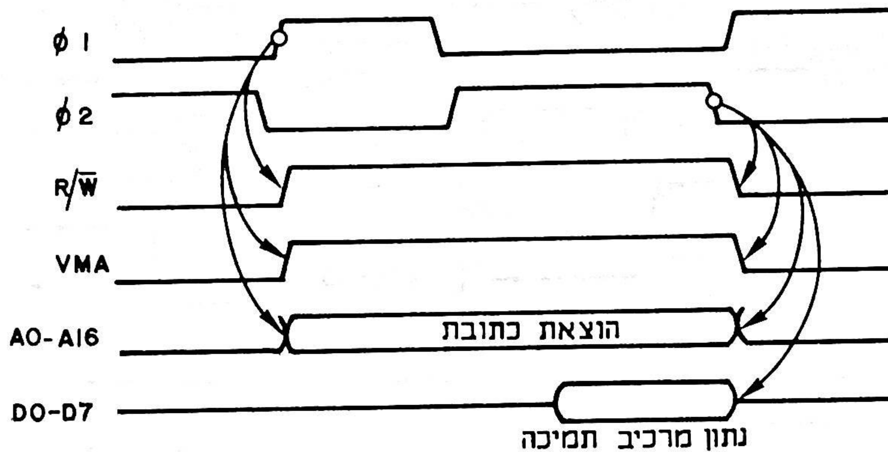
א. פעולת קריאה (READ), שבה מעביר ה-CPU בית (BYTE) של נתונים (הוראה או נתון) אליו.
ב. פעולת כתיבה (WRITE), שבה מוציא ה-CPU בית של נתונים לרכיב תמיכה.
ג. פעולה פנימית של ה-CPU, שאינה משתמשת במסלולי המערכת, כמו העברת נתון מאוגר לאוגר, פעולה לוגית, פעולה אריתמטית וכד'.

ביצוע הוראה ב-6800 דורש (בהתאם לסוג ההוראה) בין שניים לשמונה מחזורי מכונה המהווים צירופים של שלשת הסוגים שתוארו קודם. הוראות פסיקה יוצאות מכלל זה ודורשות יותר משמונה מחזורי מכונה.

למערכת החיצונית ל-CPU או לרכיב התמיכה אין זה משנה אם הבית שמושר ה-CPU מהווה לגביו הוראה או נתון מספרי כלשהו.
בכל מקרה נכנה בית זה בשם נתון.

א. מחזור מכונה בפעולת קריאה - READ MACHINE CYCLE

האותות השונים והקשר ביניהם במחזור מכונה זה מתוארים בדיאגרמה הבאה.



ציור 2 - 1 : דיאגרמת זמנים במחזור קריאה של ה-6800

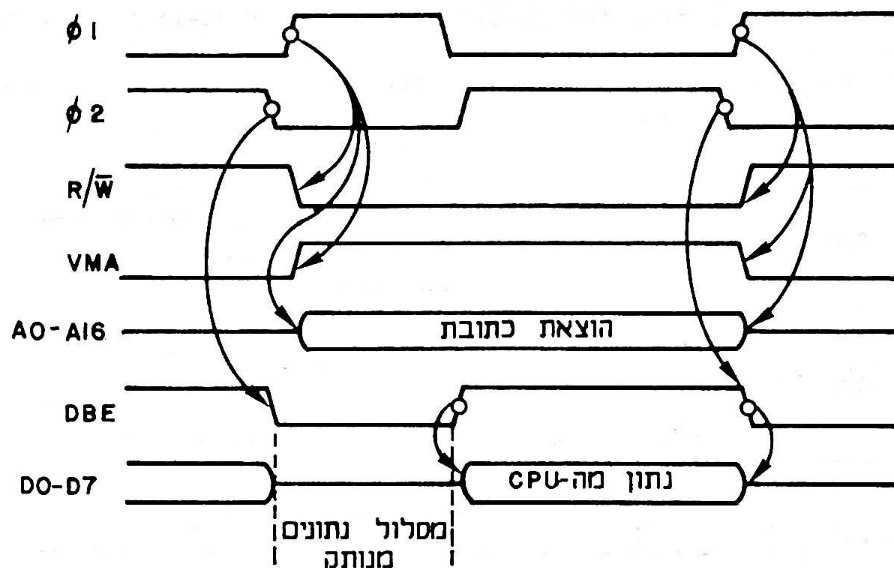
עם עליית 01, עלה קו R/W ל-HIGH. כן עולה ל-HIGH קו VMA, המציין שהכתובת שעל קוי הכתובת הינה "אמיתית" ועל מעגלי הפענוח להתייחס לכתובת זו. גם הכתובת הנדונה נמסרת לקוי מסלול הכתובות בעליית 01.

כדי לאפשר לקוי הכתובת להתייצב, מסונכרן מעגל הפענוח ל-02. זו הסבה שרכיבי התמיכה מוציאים את הנתון מהתא (שכתובתו מופיעה בקוי הכתובת) זמן מה אחרי עליית 02. עד להוצאת הנתון נמצאים קוי מסלול הנתונים במצב נתק (מצב שלישי-3-STATE).

ה-CPU אוסף אליו את הנתון עם ירידת 02. בזמן זה צריך הנתון להתייצב בקוי הנתונים.

ב. מחזור מכונה בפעולת כתיבה - WRITE MACHINE CYCLE

האותות המשתתפים במחזור מכונה זה מתוארים בציור הבא.



ציור 3 - 1 : דיאגרמת זמנים של מחזור כתיבה של ה-6800

גם כאן נמסרת הכתובת לקוי מסלול הכתובות, וקו VMA עולה ל-HIGH עם עלית 01. לעומת מחזור קריאה, קו R/W יורד ל-LOW ומציין שזהו מחזור כתיבה של ה-CPU.

ה-CPU מוציא את הנתון (אותו הוא רוצה לכתוב) זמן קצר אחרי עלית קו DBE ל-HIGH. ברוב המקרים קו DBE מקוצר ל-02 ולכן מסירת הנתון נעשית זמן קצר אחרי עלית 02.

מעגל הפענות, המסונכרן ל-02, דואג לנעילת הנתון עם ירידת 02 ל-LOW. יש לשים לב, שבזמן זה הנתון עדיין יציב בקוי הנתונים.

ג. מחזור מכונה בפעולה פנימית - INTERNAL OPERATION MACHINE CYCLE

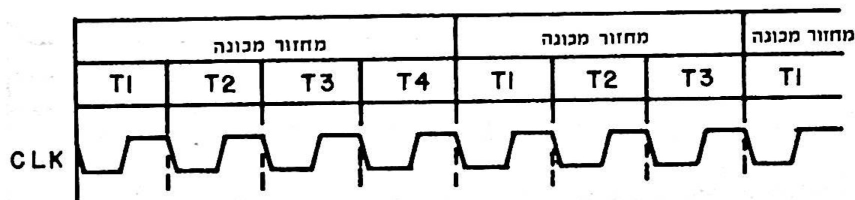
בפעולה פנימית, אין ה-CPU פונה למסלולי המערכת. המצב הלוגי של מסלול הכתובות ומסלול הנתונים לא מוגדר ואקראי. כדי לסמן זאת לרכיבי התמיכה של ה-CPU, נמצא קו VMA במצב LOW. קו R/W נמצא ב-HIGH, כדי שלא תתבצע כתיבה מקרית באחד התאים.

1.3 דיאגרמות זמנים של ה-8085A

שעון ה-8085A יוצר גל רבועי יחיד. תדר מקובל של השעון הוא 3.125 MHZ. ליצירת תדירות זו משתמשים בגביש של 6.25 MHZ, כי השעון מחלק ב-2 את התדירות. ל-8085A יש יציאת אות ברמת TTL המשמשת לסנכרון רכיבי התמיכה של המיקרופרוססור אליו. יציאה זו היא הקו CLK (CLOCK - פין 37).

ביצוע הוראה מתוזמן על ידי שורה של מחזורי מכונה, שכל אחד מורכב ממספר מחזורי שעון. ב-8085A (כמו בכל משפחת ה-8080) קיים הבדל בין מחזור מכונה למחזור שעון. מספר מחזורי השעון במחזור מכונה משתנה בהתאם לסוג מחזור המכונה ולסוג ההוראה.

מחזור המכונה הראשון של כל הוראה, יהיה בעל ארבעה או ששה מחזורי שעון. שאר מחזורי המכונה יהיו בעלי שלשה מחזורי שעון. הסבה לכך היא, שמחזור המכונה הראשון מבצע את הבאת ההוראה (INSTRUCTION FETCH). במחזור זה יחידת הבקרה צריכה לפענח את ההוראה המובאת ולבצע אותה אם היא לא כוללת פניה נוספת לזכרון. שאר מחזורי המכונה פשוטים יותר כפי שנראה בהמשך. דוגמא כללית מופיעה בציור 4-1. את מחזורי השעון מציינים האותיות T (...,T1,T2).



ציור 4 - 1 : אותות השעון במחזורי מכונה של ה-8085A.

על אף האורך המשתנה של מחזורי המכונה, אין ל-8085A יציאה מיוחדת המשמשת לזהוי התחלת מחזור מכונה חדש, אך ניתן להשתמש ביציאת ה-ALE למטרה זו. קו ה-ALE (ADDRESS LATCH ENABLE) עולה ל-HIGH בכל מחזור שעון ראשון של כל מחזור מכונה.

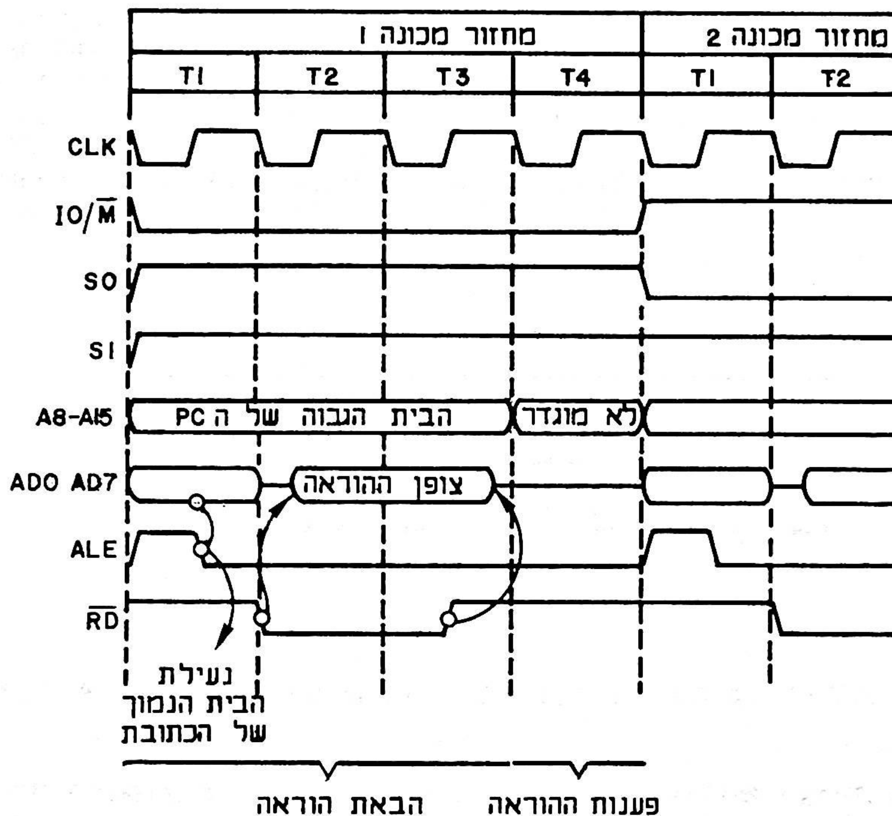
קוי AD0-AD7 משמשים הן להעברת כתובת והן להעברת נתונים בזמני שעון שונים של מחזור המכונה. במחזור השעון הראשון של מחזור המכונה, נמסרות בקוים אלה 8 הסיביות הנמוכות (A0-A7) של הכתובת הדרושה. באותו זמן מופיע דופק HIGH בקו ALE למשך חצי מחזור שעון. דופק זה נועל את 8 הסיביות הנמוכות של הכתובת ברכיב חיצוני, הכולל תאי LATCH והמיועד לכך (8212 למשל).

בשני מחזורי השעון הבאים, מוציא ה-CPU נתון או מקבל נתון על אותם קוים (AD0-AD7). הקוים A8-A15 מוציאים, בכל שלשת מחזורי השעון הראשונים, את 8 הסיביות הגבוהות של הכתובת. מכאן רואים, שלצורך קבלת נתון או מסירת נתון על ידי ה-CPU, דרושים 3 מחזורי שעון.

בכל מחזור מכונה ב-8085A (מלבד במחזורים של BUS פנוי) קיימת הבאת נתון או מסירת נתון. לכן המספר המינימלי של מחזורי השעון במחזור מכונה הוא 3.

ל-8085A מספר סוגים רב יחסית של מחזורי מכונה. נתאר את אלה הקשורים בזכרון.

א. מחזור מכונה ראשון - הבאת הוראה (FETCH INSTRUCTION)



ציור 5 - 1 : דיאגרמת זמנים של מחזור הבאת הוראה ב-8085A

בדיאגרמת זמנים זו יש לשים לב למספר פרטים

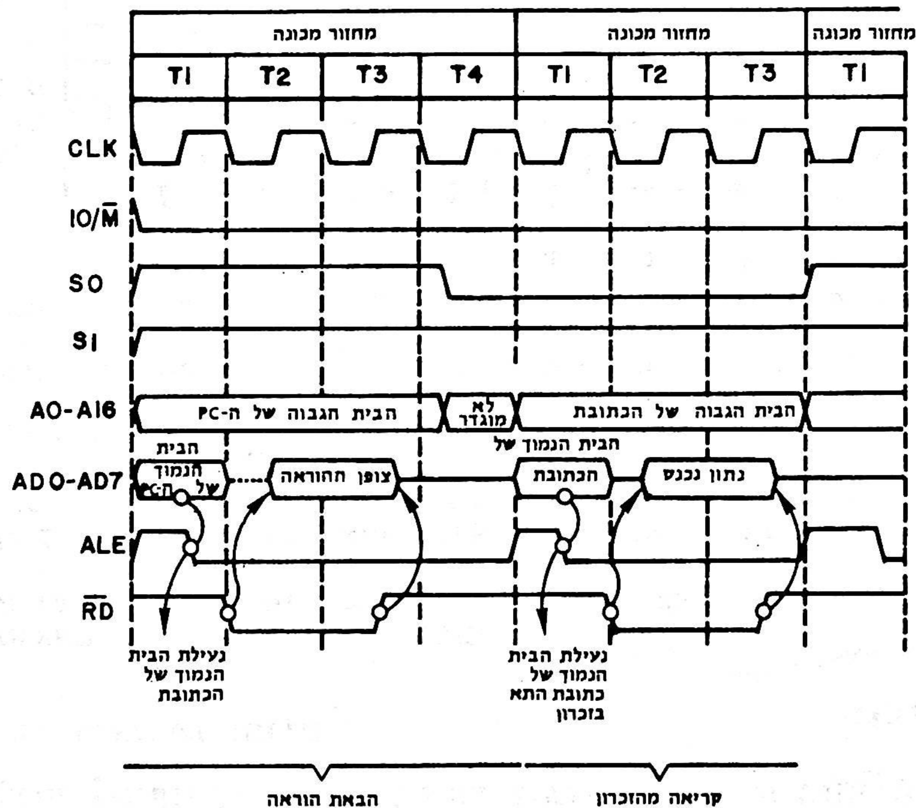
1) השלב שבו נועל קו ALE את הבית (BYTE) הנמוך של הכתובת, הוא בירידה שלו ל- LOW. משך השהייה שלו ב-HIGH מאפשר לקוי הכתובת להתייצב.
2) הכתובת במחזור מכונה זה של הבאת ההוראה מגיעה ממונה התוכנית (PC).
3) קו IO/M הוא ב-LOW,כלומר, הפניה היא אל הזכרון לצורך הבאת צופו ההוראה (OBJECT CODE).
4) קוים S0 ו-S1, המשמשים לזהוי סוג מחזור המכונה, נמצאים ב-HIGH, כדי לסמן מחזור קריאה מהזכרון.
5) קו RD נמצא ב-LOW בזמן העברת הנתון בלבד. קליטת הנתון על ידי ה-CPU נעשית עם עלית קו RD ל-HIGH.

שאר מחזורי השעון של מחזור המכונה הראשון (אחד או שלשה) משמשים את ה-CPU לפענוח ההוראה ולביצועה. בציור מתואר מחזור מכונה ראשון בן 4 מחזורי שעון. אם ההוראה המובאת אינה דורשת פניה נוספת לזכרון (כמו במיעון מובלע), אזי מחזור המכונה יהיה בן ששה מחזורי שעון. המחזורים החמישי והששי מיועדים לבצוע ההוראה. בזמן מחזורי השעון הרביעי עד הששי, מנותקים הקוים AD0-AD7 והקיים A8-A15 לא מוגדרים. נקודה זו צריכה להילקח בחשבון בזמן פענוח הכתובת.

אם ההוראה דורשת פניה נוספת לזכרון, יהיה מחזור המכונה הראשון בן ארבעה מחזורי שעון.

ב. מחזור קריאה מהזכרון (READ MACHINE CYCLE).

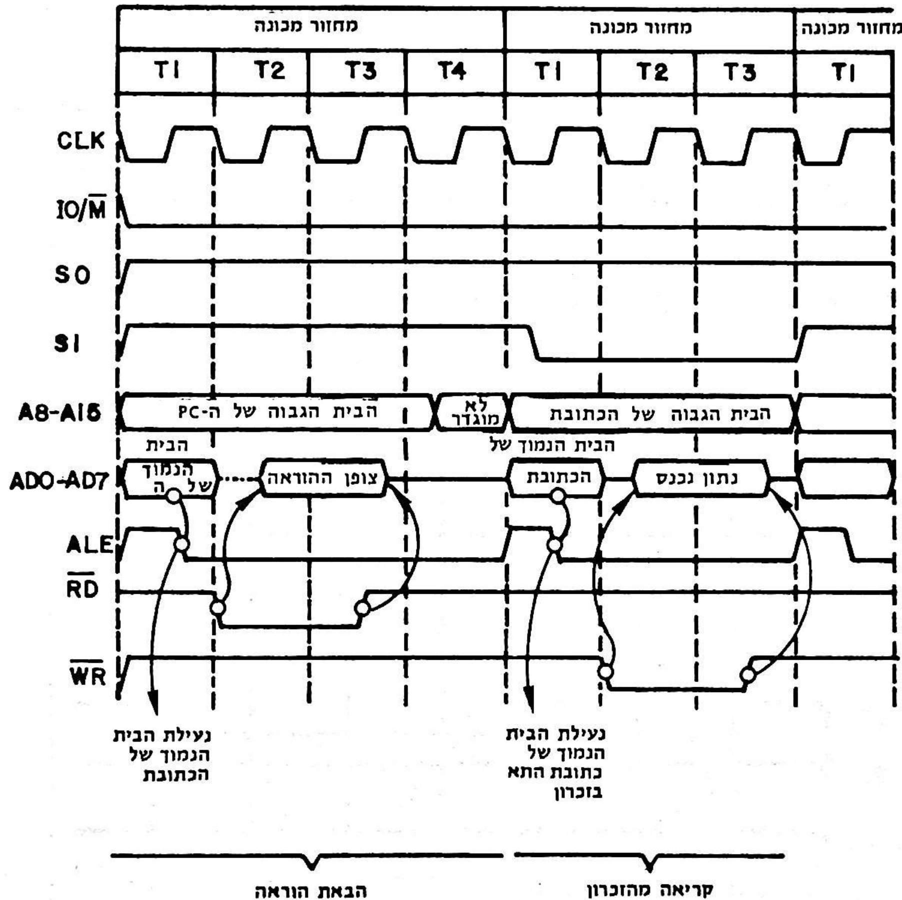
בציור הבא מתוארת דיאגרמת הזמנים של קריאת נתון מהזכרון, הבאה לאחר מחזור מכונה ראשון של הבאת ההוראה. MC1 (מחזור מכונה ראשון - MACHINE CYCLE 1 -) הוא בן 4 מחזורי שעון, מכיוון שההוראה כוללת מחזור מכונה נוסף. מחזור מכונה זה (MC2) הוא בן שלשה מחזורי שעון והוא דומה לשלשת מחזורי השעון הראשונים של MC1, מלבד קוים S0 ו-S1. S0 נמצא ב-LOW ו-S1 נמצא ב-HIGH.



ציור 6-1: דיאגרמת זמנים של מחזור קריאה מהזכרון ב-8085A.

ג. מחזור כתיבה בזכרון (WRITE MACHINE CYCLE)

מחזור MC2 המתואר בציור הבא, מבצע כתיבה בזכרון לאחר הבאת ההוראה ב-MC1. במחזור זה קו WR יורד ל-LOW לצורך בצוע הכתיבה. קו SO נמצא ב-HIGH וקו S1 ב-LOW. קו IO/M נמצא ב-LOW כדי לציין פניה לזכרון.



ציור 7 - 1 : דיאגרמת זמנים של מחזור כתיבה בזכרון ב-8085A.

ל-8085A יש סוגים נוספים של מחזורי מכונה אבל לא נדון בהם פה. ניתן למוצאם בדפי הנתונים של הרכיב.

1.4 זיהוי דיאגרמת זמנים

תנאי הכרחי לאיתור תקלות הוא זהוי והכרת דיאגרמות הזמנים של ה-CPU. איתור תקלות בחומרה במערכת מיקרו-מחשב קשה, מכיוון שהמיקרו-מחשב הוא מערכת דינאמית. קשה להסתכל על פולסי השעון שלו ולזהות את הבעיה בעזרת משקף תנודות (אלקטרוסקופ). קיימים אמצעים מתוחכמים ויקרים לאיתור תקלות, אבל לא נעסוק בהם. בעיה נפוצה היא, שבפניה לכתובת מסוימת, אין ה-CPU מקבל את הנתון הדרוש, או הנתון שהוא מוסר אינו מגיע ליעדו מסבה כלשהי.

השיטה באיתור תקלה מסוג זה היא להריץ את ה-CPU בלולאה אינסופית ובתוך הלולאה הוא פונה פעם אחת לכתובת הבעייתית. קשה מאד להסתנכרן על פולסי השעון ולזהות את מחזור המכונה האמור. אם קיים מעגל פענוח לכתובת הבעייתית, קל להסתנכרן ליציאתו. קו היציאה מגיב רק בפניה לכתובת הנ"ל וזה קורה פעם אחת בכל לולאה. עם ההסתנכרנות, בודקים בעזרת הערוץ השני של המשקף, את מצב קוי הנתונים וקוי הכתובת של הרכיב המחובר.

בשיטה זו של איתור תקלות נטפל בפרק הבא. בפרק זה, כדי לזהות דיאגרמת זמנים של ה-CPU, נשתמש בשיטה הבאה. נריץ תוכנית שבה ה-CPU פונה לכתובת מסויימת, פעם אחת במחזור. כתובת זו תהיה שונה מכל שאר הכתובות, שאליהן פונה ה-CPU בביצוע התוכנית, לפחות בסיבית אחת. לדוגמא, התוכנית כתובה בתחום כתובות נמוך מ-8000H והכתובת המסוימת היא 8050. רק במחזור המכונה הפונה לכתובת זו עולה קו A15 ל-HIGH. מכיוון שזה קורה פעם אחת בכל לולאה ניתן להסתנכרן על קו זה.

ב-6800 נריץ את התוכנית הבאה:

```
0010    86 55       START:    LDAA#$55
0012    B7 80 50    LOOP:     STAA $8050
0015    7E 00 12              JMP  LOOP
```

תוכנית זו, לאחר ביצוע הפקודה הראשונה, רצה על תחום הכתובות 0012 עד 0017 בלולאה אינסופית. במהלך ביצוע לולאה זו, פונה התוכנית פעם אחת לכתובת 8050 ומעבירה אליה את הנתון 55. בפניה זו עולה קו A15 ל-HIGH ומאפשר הסתנכרנות על מחזור מכונה זה.

ב-8085A נריץ תוכנית דומה לתוכנית הבאה:

```
2000    3E 55       START:    MVI A,55H
2002    32 50 80    LOOP:     STA 8050H
2005    C3 02 20              JMP LOOP
```

תוכנית זו, לאחר ביצוע הפקודה הראשונה, רצה על תחום הכתובות 2002 עד 2007 בלולאה אינסופית. במהלך ביצוע לולאה זו, פונה התוכנית פעם אחת לכתובת 8050 ומעבירה אליה את הנתון 55. בפניה זו עולה קו A15 ל-HIGH ומאפשר הסתנכרנות על מחזור מכונה זה.

1.5 שאלות הכנה

1) שרטט דיאגרמת זמנים מפורטת, הכוללת את אותות שעון ה-CPU, קוי הבקרה, 8 קוי הנתונים ו-16 קוי הכתובת, שמתארת את מצב הקוים כאשר ה-CPU כותב את הנתון AA בכתובת 2155.

2) שרטט דיאגרמת זמנים מפורטת, כמו בשאלה 1), שמתארת את מצב הקוים כאשר ה-CPU קורא נתון מהכתובת A265. בכתובת זו נמצא נתון 6E.

מהלך הניסוי

1.6 זהוי דיאגרמת זמנים

מטרת הניסוי לקרוא מצב קוי המסלולים במחזורי המכונה השונים, ולזהות דיאגרמות זמנים של המערכת.

1) הרץ את התוכנית הרשומה בסעיף 1.4 בהתאם לסוג המיקרופרוססור שבו אתה משתמש.

2) חבר את הבחון (PROB) של ערוץ Y1 לקו A15. כוון את כפתור ה-TIME BASE וכפתור ה - LEVEL כך שתסתנכרן על אות זה. בדוק שהמשקף מכוון על הערוץ המתאים ובמצב .D.C.

3) חבר את בחון ערוץ Y2 לקו אתות השעון של ה-CPU (02 ב-6800 או CLK ב-8085A). שרטט את אותות השעון הנראים בתחום של שני מחזורים של A15. שרטט מתחת לאותות השעון את האותות של A15 בהתאם.

4) עבור עם בחון ערוץ Y2 לקו A14 ושרטט בדייקנות את האותות המופיע בו בהתאמה ל-A15. הפוך את הגל לגל מרובע והתעלם מהרעשים שמופיעים על הקו.

5) עבור בצורה דומה על הקוים A13-A0 ו - D7-D0.

6) עבור על קוי הבקרה הדרושים. VMA ו - R/W ב-6800. IO/M, RD, WR ו-ALE ב-8085A.

7) זהה וסמן את מחזורי המכונה בדיאגרמת הזמנים שקבלת. רשום לכל מחזור מכונה את הכתובת שאליה הוא פונה, הנתון המועבר וכן האם זה מחזור כתיבה או קריאה.

פרק 2 - פענוח כתובות

2. נושאי הפרק

א. אופן קביעת רכיבי זכרון RAM ו-ROM בתחומי כתובות מוגדרים.
ב. אופן בנית מערך יחידות זכרון בעזרת מפענחים (DECODERS).
ג. איתור תקלות בחומרה.
ד. אופן חיבור חוצצים ל-BUS.

רקע עיוני

2.2 קביעת רכיבי זכרון RAM בתחומי כתובות מוגדרים

יחידות הזכרון מתחברות אל מסלולי ה-CPU - מסלול הכתובות, מסלול הנתונים ואל מספר קוי בקרה. ב-6800, מסלול הכתובות נפרד ממסלול הנתונים וכולל 16 קוי כתובת. ב-8085, חצי ממסלול הכתובות הוא מסלול מרובב (MULTIPLEXED BUS). בקוים (AD0-AD7) מוציא ה-CPU כתובת בחלק הראשון של מחזור המכונה, ובחלק השני זורמים בו נתונים. כדי לקבל כתובת מלאה של 16 קוים לאורך כל מחזור המכונה, נועלים את הכתובת שנמסרה בקוים AD0-AD7 בעזרת קו ALE, ברכיב הכולל תאי LATCH, כמתואר בפרק הקודם. בפרק זה ובפרקים הבאים, נתייחס אל מסלולי ה-8085 לאחר הפרדתם, כלומר, 8 קוי נתונים ו-16 קוי כתובת בנפרד.

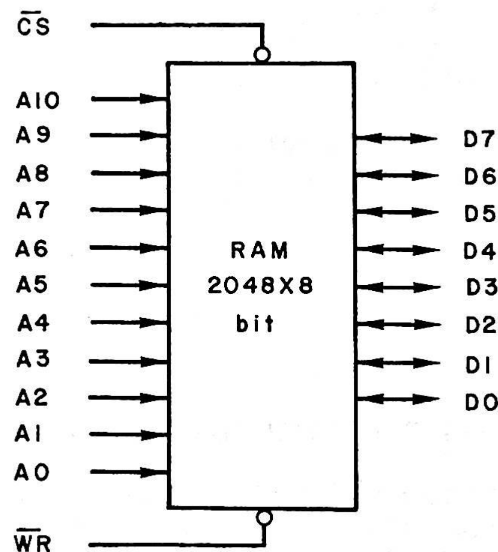
ליחידת הזכרון המתחברת, נתייחס כאל יחידת זכרון אחת בעלת מספר תאים מוגדר וכל תא הוא בן 8 סיביות. יחידה כזו יכולה להיות מורכבת מכמה תת יחידות.

ליחידת זכרון כזו יש 8 קוי נתונים ו-N קוי כתובת בהתאם למספר התאים. מספר התאים שווה ל-2 בחזקת N. לכל יחידת זכרון יש קו CS (CHIP SALECT), המאפשר ניתוק התאים מקוי הנתונים. כאשר קו זה הוא ב-LOW, מחובר התא שכתובתו מצוינת בקוי הכתובת אל קוי הנתונים של הרכיב (ראה ציור 1-1). כאשר קו זה הוא ב-HIGH, גם אם רשומה כתובת בקוי הכתובת, מנותקים התאים מקוי הנתונים. קוי הנתונים של הרכיב נמצאים במצב נתק הנקרא מצב של התנגדות גבוהה - HIGH TMPEDANCE או מצב שלישי - 3-STATE (לא אפס לוגי ולא אחד לוגי).

ליחידות ה-RAM קו נוסף WR הקובע את כיוון זרימת הנתונים. במצב LOW, זורמים הנתונים אל ה-RAM. זהו מצב WRITE (ה-CPU כותב בו). במצב HIGH זורמים הנתונים מה-RAM אל ה-CPU. זהו מצב READ (ה-CPU קורא ממנו).

לחלק מיחידות הזכרון יש קו נוסף לקו ה-CS הנקרא OE (או OD). קו זה מיועד לצורות חיבור נוספות המקטינות את צריכת הזרם של הרכיב. בחיבור רגיל מחובר קו זה לקו ה-CS.

נתאר יחידת זכרון RAM בת 11 קוי כתובת (2K תאים).



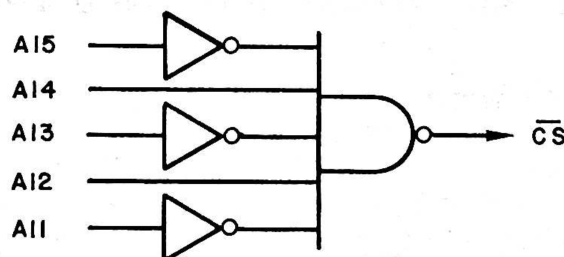
ציור 2 - 1 : 2KX8 BIT RAM

המספר הנרשם בקוי הכתובת המחוברים לרכיב קובע איזה תא מבין תאי הרכיב מתחבר לקוי נתונים. קוים אלה מתחברים ישירות אל הקוים המקבילים ב-CPU. כדי לקבוע את יחידת הזכרון בתחום כתובות מסוים, מפענחים בעזרת מעגל פענוח את סיביות ה-M.S.B שאינן מחוברות אליה.

נמקם את יחידת הזכרון שמופיעה בציור הקודם בתחום הכתובות 5000-75FF. בתחום זה יש 2048 כתובות. בדוק זאת. כל פניה לכתובת בתחום כתובות זה תיצור צירוף כלשהו בקוים A0-A10 אבל הקוים A11-A15 יציגו את המספר הבינארי 01010.

	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
5000H=	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
כתובת בתחום=	0	1	0	1	0	X	X	X	X	X	X	X	X	X	X	X
57FFH=	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1

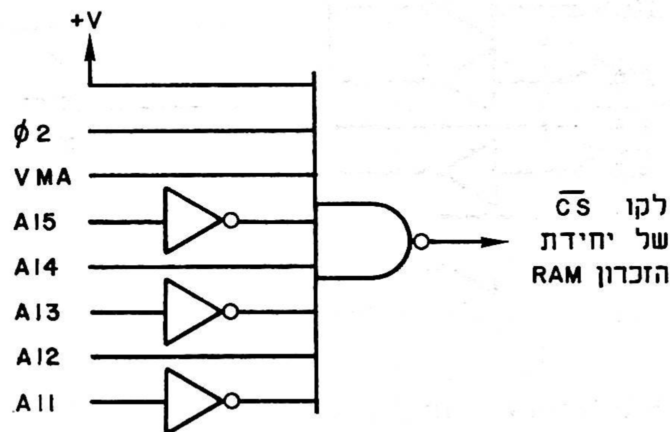
מעגל הפענוח יפענח את הקוים A11-A15. יציאתו תרד ל-LOW כאשר בכניסותיו יופיע הצירוף 01010. מעגל כזה יראה בצורה הבאה:



ציור 2 - 2 : מעגל פענוח לזיהוי 01010

יחידת הזכרון המחוברת לקו CS תמותג רק כאשר יתקבל בקוים A11-A15 הצירוף המבוקש (01010). למעשה אין די בפענוח הכתובת המתאימה. את ירידת CS יש להתנות גם במצבם של קוי הבקרה.

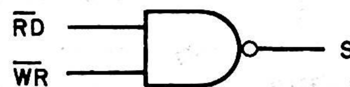
ב-6800 יש להוסיף למפענח את הקוים VMA (כדי שיפענח רק כתובת אמיתית) ו-02 כדי ש-CS יתקבל רק לאחר התייצבות קוי הכתובת. מעגל שלם מתואר בציור 3-2.



ציור 3 - 2 : מעגל פענוח מלא ל-RAM ב-6800

בדוק באיזה מצב של הכניסות ירד קו ה-C̄S ל-LOW.

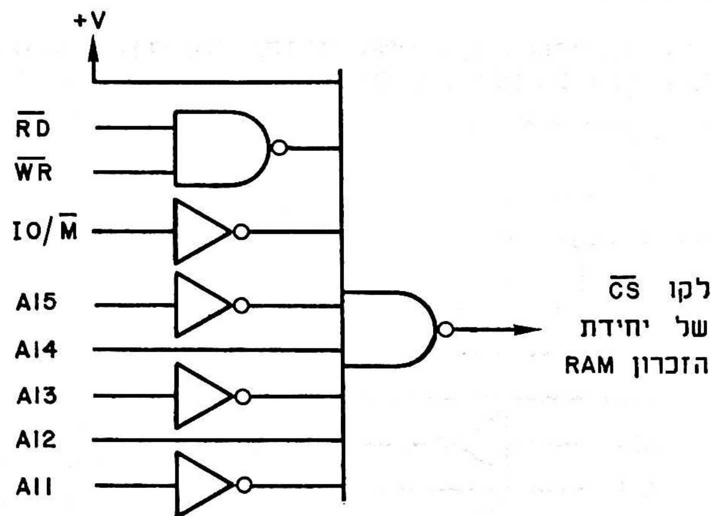
ב-8085 אין קו VMA אבל כתובת אמיתית מתקבלת רק כאשר WR או RD ב-LOW. לכן ניתן ליצור קו כזה בעזרת שער NAND.



ציור 4 - 2 : מעגל ליצירת קו VMA ב-8085

קו S יעלה ל-HIGH רק כאשר RD או WR ירדו ל-LOW. קו זה מספק גם את הסנכרון למערכת, כלומר, יגרום ש-CS יתקבל רק לאחר התייצבות קוי הכתובת. במעגל זה יש להוסיף פענוח של קו IO/M כדי להבחין בין רכיב זכרון לרכיב קלט/פלט.

מעגל הפענוח ב-8085 יראה בצורה הבאה:



ציור 5 - 2 : מעגל פענוח מלא ל-RAM ב-8085

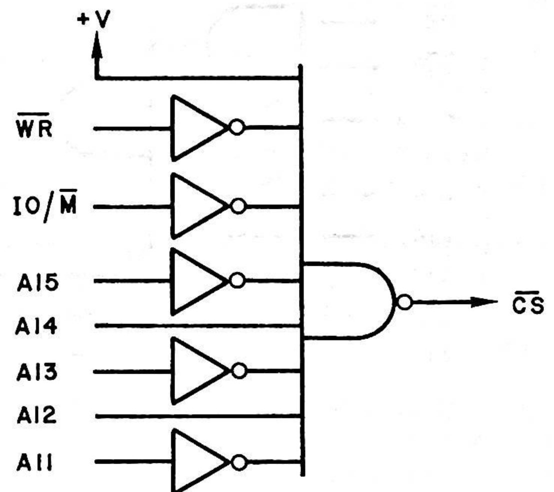
בדוק באיזה מצבים של הכניסות ירד קו CS ל-LOW.

ב-6800 מחברים לקו WR ב-RAM את קו R/W של ה-CPU, כדי לקבוע את כיוון זרימת הנתונים ברכיב.

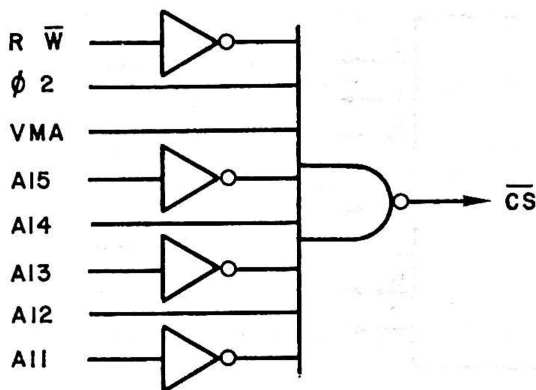
ב-8085 מחברים לקו WR ב-RAM את קו WR של ה-CPU.

2.3 קביעת יחידת ROM בתחום כתובות מוגדר

יחידת ROM בנויה כמו יחידת RAM, רק ללא קו WR, מכיוון שכיוון זרימת הנתונים הוא רק מהרכיב אל ה-CPU. לכן אין מחברים אליה את הקוים R/W או WR. לעומת זאת רוצים שקו CS ירד ל-LOW רק במצב קריאה מהרכיב. מעגלי הפענוח כמו בציורים הבאים.



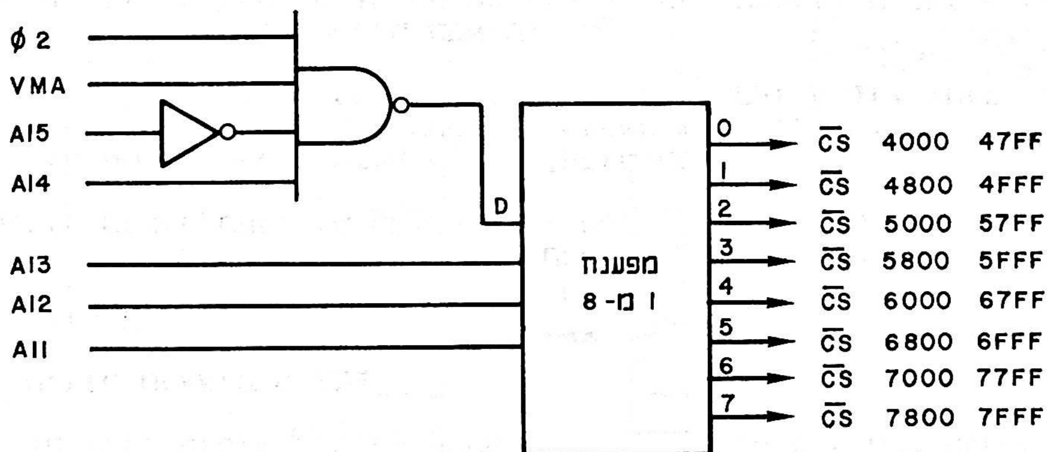
ציור 6-2: מעגל פענוח ל-ROM ב-8085



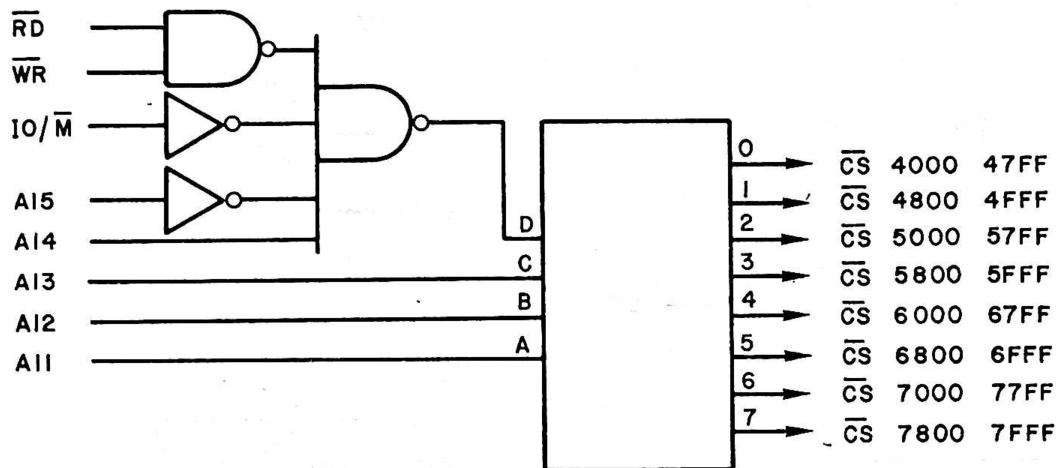
ציור 7-2: מעגל פענוח ל-ROM ב-6800

2.4 בנית מערך יחידות זכרון בעזרת מפענחים (DECODERS)

את מעגל הפענוח ניתן לממש בצורות שונות. אחת השיטות לכך היא בעזרת מפענחים. בשיטה זו מקבלים למעשה מספר קוי CS, כאשר כל אחד הוא עבור תחום כתובות אחר. חיבור יחידות זכרון לקוי CS אלה יוצר מערך תאי זכרון על תחום כתובות רחב.



ציור 8 - 2 : מעגל פענוח ל-8 יחידות זכרון ב-6800



ציור 9 - 2 : מעגל פענוח ל-8 יחידות זכרון ב-8085

2.5 איתור תקלות בחומרה

כפי שנאמר בפרק הקודם, השיטה באיתור תקלה בחומרה, היא הרצת ה-CPU בלולאה אינסופית כאשר בתוך הלולאה הוא פונה פעם אחת לכתובת הבעייתית. יציאת מעגל הפענוח מספקת אות סנכרון המסמן מתי ה-CPU פונה לכתובת זו.

קו CS של מעגל הפענוח ימצא ב-LOW רק כאשר תופיע בקוי הכתובת כתובת מתאימה וקוי הבקרה ימצאו במצב מתאים. בכל כתובת אחרת ימצא קו CS ב-HIGH. כדי לבדוק את זרימת האותות על קוי המסלולים בפניה לכתובת מסוימת, יש לבצע את השלבים הבאים:

א. הרצת תוכנית הפונה במהלכה פעם אחת בלבד לכתובת זו. הדבר ייצור את מחזורי המופיע פעם אחת בכל לולאה של התוכנית.
ב. חיבור אחד מבחוני המשקף ליציאת מעגל הפענוח והסתנכרנות על אות זה.
ג. חיבור הבחון השני של המשקף לקוי המסלולים בזה אחר זה ובדיקת מצבם בהתאמה לאות המוצא של קו הפענוח. ניתן להשוות בין האותות המופיעים על רגלי ה-CPU לאותות המופיעים על הרגלים המתאימות של יחידות הזכרון.

2.6 חיבור חוצצים ל-BUS

נושא זה אינו מופיע כתרגיל במהלך הניסוי. אבל בין CPU מערכת המיקרו-מחשב לערכת התרגול נמצאים חוצצים ולהם חשיבות רבה. לכן הוכנס סעיף זה לרקע העיוני.

כאשר מחברים מספר רב של רכיבי תמיכה למסלולי ה-CPU, יוצר הדבר העמסה על קוי ה-CPU. כדי להתגבר על בעיה זו, יש לחבר חוצצים בין יציאות ה-CPU למסלולים.

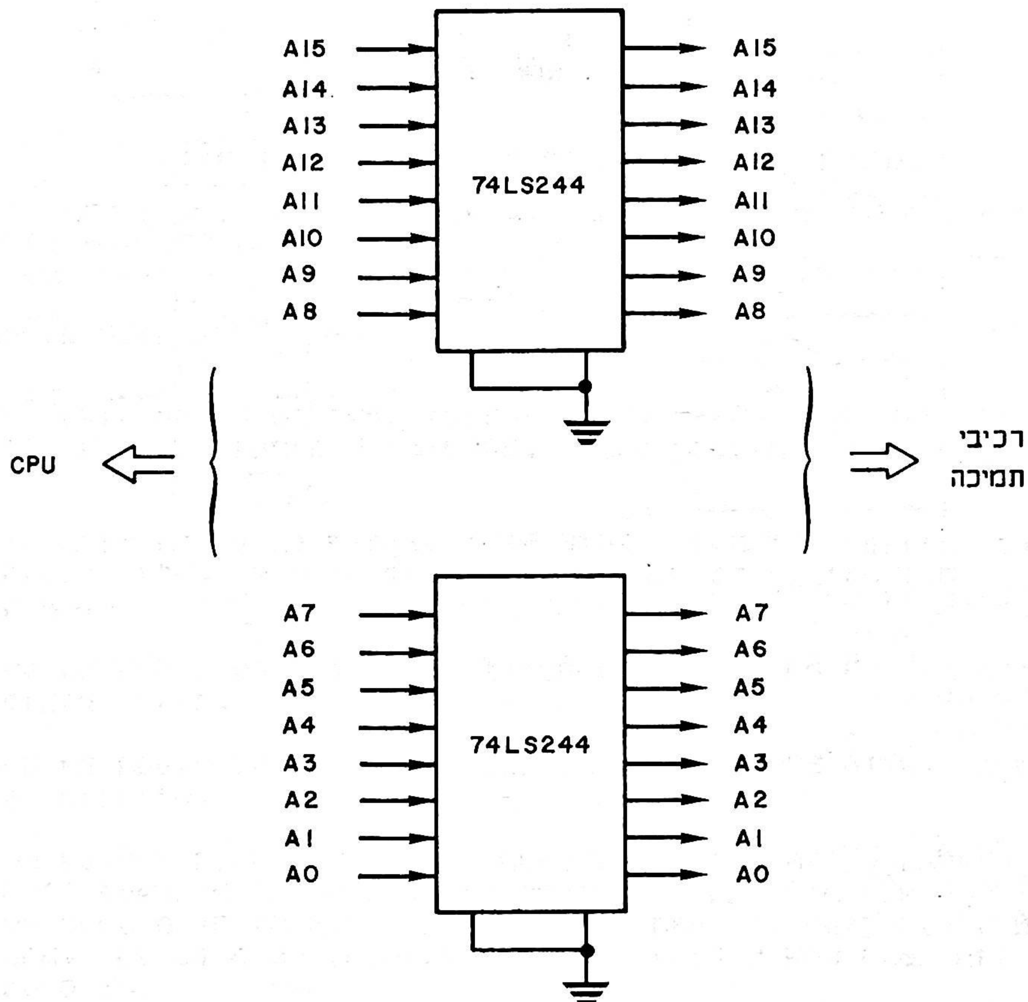
תפקידי החוצצים:

1) לאפשר צריכת זרם גבוהה יותר מאשר יכול לספק ה-CPU והגדלת ה-FAN OUT שלו.
2) הפרדה בין ה-CPU לרכיבי התמיכה, כך שתקלה בחיבור הרכיבים לא תגרום נזק ל-CPU.
3) ערכת המוצא הנמוכה של החוצצים מקטינה את הרעשים על הקוים השונים.

קיימים שני סוגי חוצצים - חד כיווניים לקוי הכתובת והבקרה, ודו-כיווניים - לקוי הנתונים. לחוצצים יש בדרך כלל קו CS, המאפשר ניתוקם במידת הצורך. במקרה של שימוש בהם לצורך הגדלת ה-FAN OUT, אפשר לחבר קו זה באופן קבוע לאדמה.

חוצצים מקובלים הם ה-74LS244 כרכיב חד כיווני לקוי כתובת, ה-74LS365 כרכיב חד כיווני לקוי בקרה (מכיל 6 חוצצים) וה-74LS245 כרכיב דו כיווני לקוי הנתונים.

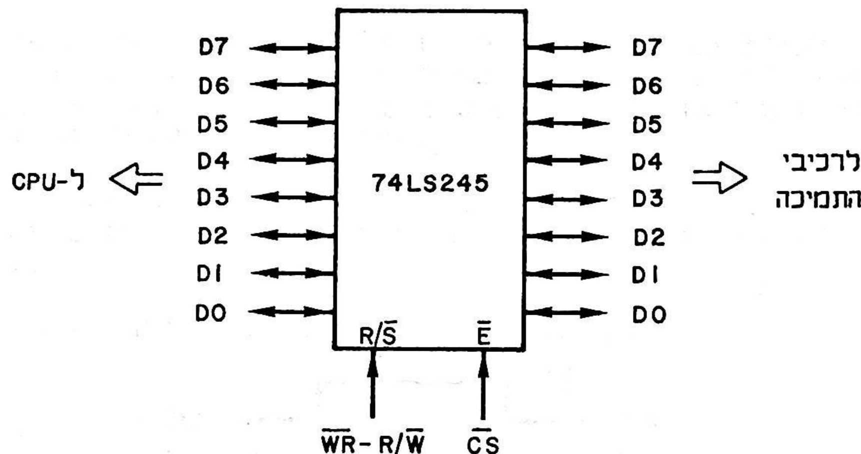
חיבור ה-74LS244 לקוי הכתובת יעשה בצורה הבאה:



ציור 10 - 2 : חיבור חוצצים לקוי הכתובת

חיבור ה-74LS365 לקוי הבקרה נעשה בצורה דומה לחיבור ה-74LS244.

בחיבור ה-74LS245 לקוי הנתונים, מחברים את קו R/W (ב-6800) או קו WR (ב-8085)מלכניסת הכיוון (SEND/RECEIVE - S/R) של הרכיב. כניסה זו קובעת את כיוון העברת הנתון ברכיב. את כניסת ה-CS של הרכיב ניתן לחבר באופן קבוע לאדמה. החיבור מתואר בציור 2-11.



ציור 11 - 2 : חיבור חוצצים לקוי הנתונים.

אפשרות אחרת היא להחליף בין A ל-B. במקרה כזה מחברים לקו R/S את קו RD או R/W דרך מהפך.

2.7 שאלות הכנה

1) תכנן מעגל פענוח שיציאתו (קוCS) יורדת ל-LOW כאשר קוי הכתובת מציינים כתובת בתחום 8000-87FF. מעגל הפענוח צריך לפענח גם את קוי הבקרה.

2) תכנן מעגל הקובע את ה-6116 (2KX8 RAM - ראה דפי נתונים) לתחום הכתובות הנ"ל. שים לב ש-CS צריך להתקבל גם בכתיבה וגם בקריאה.

3) שכלל את המעגל שתכננת בסעיף 1) כך שיתקבל CS רק בקריאה מתחום הכתובות שצויין.

4) השלם את המעגל על ידי חיבור לקו זה יחידת 2716 EPROM. היעזר בדפי הנתונים.

5) שרטט מעגל זכרון של 16K RAM, המשתמש ביחידות 6116, במפענח 74LS42 ובשערים. תחום הכתובות של כרטיס זכרון זה הוא 8000-BFFF. הוסף לכרטיס זה גם מערכת חוצצים לקוי הכתובת, לקוי הנתונים ולקוי הבקרה. כניסת ה-CS של חוצץ הנתונים יורדת ל-LOW כאשר פונים לכרטיס זה.

6) כתוב תוכנית המאחסנת את המספר 80H בתא 8010H בלולאה אינסופית.

מהלך הניסוי

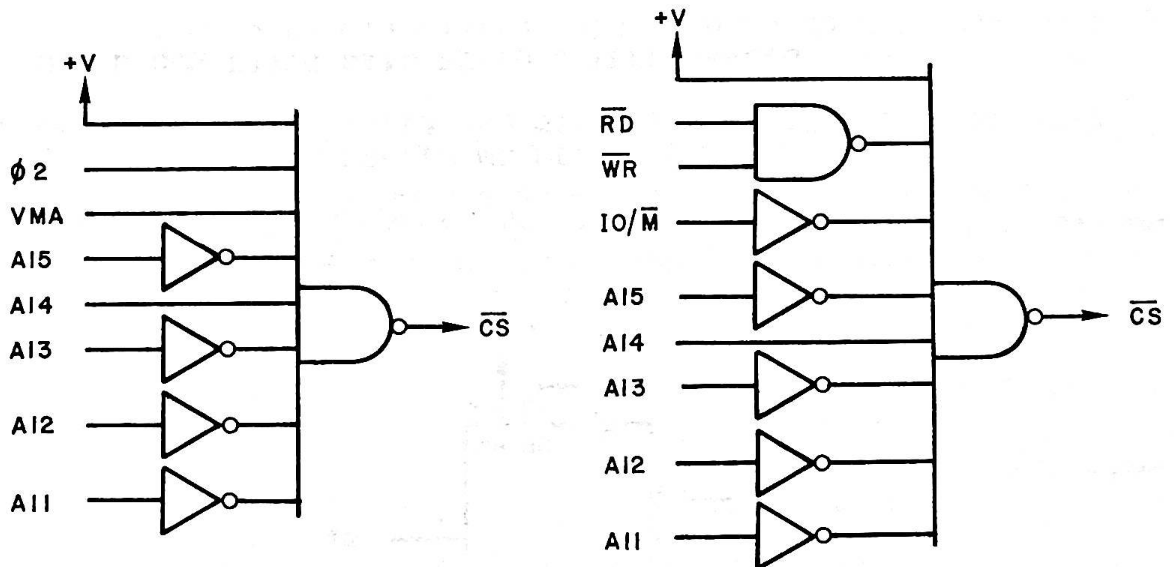
2.8 בנית CS בעזרת מעגל פענוח המורכב משערים

מטרת הניסוי היא בנית מעגל פענוח לתחום כתובות מסוים והתבוננות באותות יציאתו כאשר ה-CPU פונה לתחום כתובות זה.

1) בנה מעגל פענוח על מטריצת החיבורים בעזרת שערים לתחום הכתובות 4000-47FF

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	1	0	0	0	X	X	X	X	X	X	X	X	X	X	X

מעגל הפענוח צריך לפענח את הקוים A11-A15 ואת קוי הבקרה.
ההתחברות לקוים היא דרך נקודות החיבור שבמטריצה העליונה D23 - D60



ציור 2-12: מעגלי פענוח ל-8085 ול-6800 לחיבור RAM.

2) כתוב את התוכנית הבאה הפונה פעם אחת בלולאה לכתובת בתחום הכתובות המוגדר.

6800

```
START:  LDAA#$0F
LOOP1:  STAA $4010
        BRA  LOOP1
```

8085

```
START:  MVI  A,0FH
LOOP1:  STA  4010A
        JMP  LOOP1
```

3) חבר את הבחוו של ערוץ Y1 במשקף התנודות ליציאת ה-CS של המעגל. הרץ את התוכנית והסתכלו על Y1. בדוק קבלת דפקים בערוץ זה.

4) אם אינך מקבל דפקים בקו ה-CS, חבר את ערוץ Y1 לקו R/W (ב-6800)
או לקו WR (ב-8085). גם כאן תקבל דופק אחד בלוגיקה שלילית בכל
לולאה (מדוע?). הסתנכרן עליו.
בעזרת ערוץ Y2 בדוק את האותות המגיעים לכניסות שער ה-NAND בזמן
הדפקים המופיעים ב-Y1.
אם לא קבלת דפקים ב-Y1, בדוק את התוכנית או את חיבורי המערכות.

5) לאחר ההסתנכרנות עם ערוץ Y1 על קו ה-CS, בדוק בעזרת ערוץ Y2
במקביל ל-Y1 את מצב הקוים הבאים בזמן קבלת CS.

ב - 6800 - D0-D7, R/W, VMA, A0-A15, 02, CS.
ב - 8085 - D0-D7, WR, RD, IO/M, A0-A15, CK, CS.

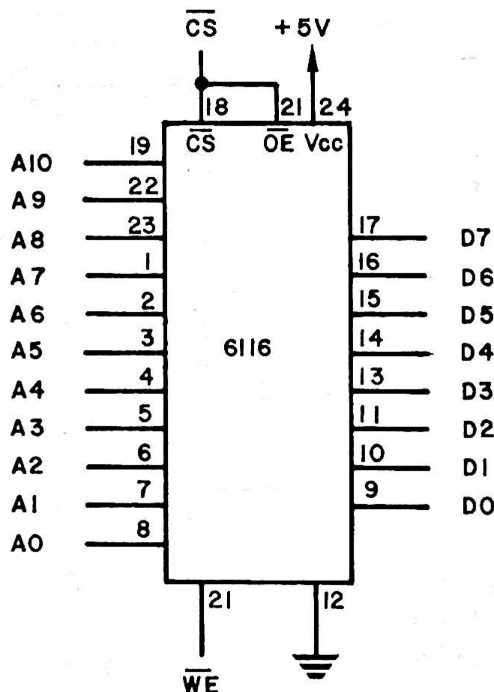
רשום מצב הקוים בטבלה.

6) זהה את הכתובת והנתון שמוציא ה-CPU בזמן קבלת אות ה-CS.
הכתובת צריכה להיות 4010 והנתון צריך להיות OF.

2.9 חיבור RAM בעזרת מעגל פענוח המורכב משערים

מטרת הניסוי היא לחבר יחידת RAM 6116 (2KX8 BIT), לתחום הכתובות
4000-47FF בעזרת מעגל הפענוח של הניסוי הקודם.

1) חבר RAM 6116 למעגל הפענוח של סעיף 2.8 כמתואר בציור 13-2.



ציור 13 - 2: חיבור 6116 - 2KX8 BIT RAM

2) פנה לכתובת 4000 והכנס לשם את המספר 00.

3) עבור לכתובת 4001 והכנס לשם את המספר 01.

4) המשך עד לכתובת 4009 והכנס את המספרים עד 09.

5) חזור לכתובת 4000 ובדוק אם הנתונים אכן נמצאים שם. אם כן, סימן שחיבוריך תקינים.

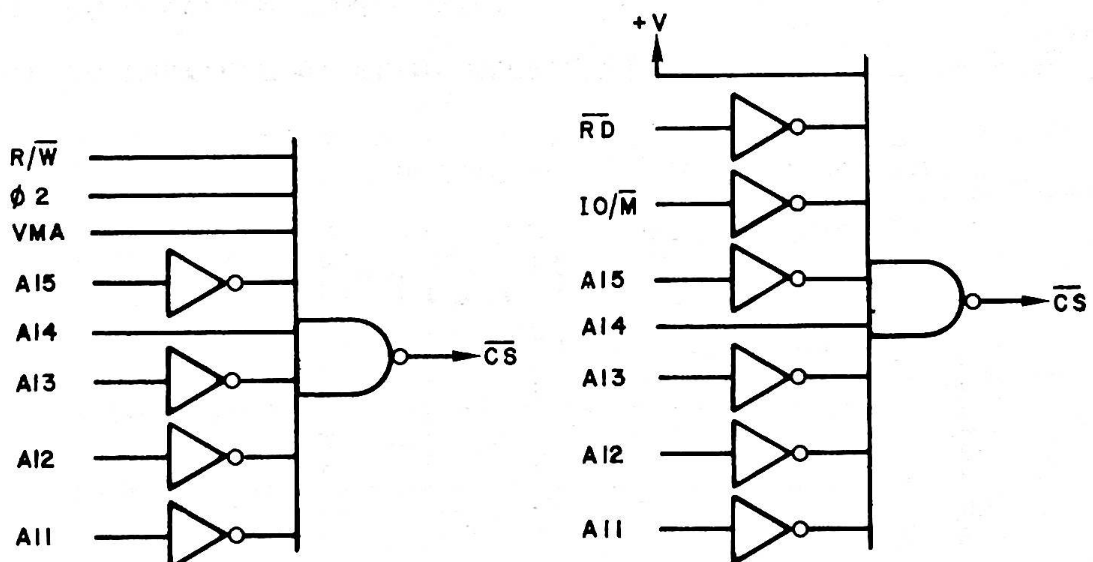
6) איתור תקלה בחיבור ה-RAM.
א) הרץ את התוכנית שבסעיף 2.8 ובדוק קבלת CS ברגליים 18 ו-21 של ה-6116.
ב) אם קבלת CS סימן שמעגל הפיענוח תקין. אם לא קבלת, בדוק את מעגל הפענוח לפי סעיף 2.8 - 4).
ג) הסתנכרן בעזרת ערוץ Y1 אל קו ה-CS. בדוק קבלת כתובת ברגלי הכתובת של ה- RAM, קבלת W בקו WE ונתון מתאים בקוי הנתונים של הרכיב.

7) שרטט מעגל שלם הכולל CPU, מעגל הפענוח ויחידת ה-RAM. הסבר את המעגל.

2.10 חיבור ROM בעזרת מעגל פענוח המורכב משערים

בחיבור ROM מעוניינים ש-CS יתקבל רק בקריאה מהרכיב ולא בכתיבה בו, אם כי אין הכרח בכך. מטרת הניסוי היא לחבר יחידת EPROM מסוג 2716 (2KX8 BIT) לתחום הכתובות אליו חברנו את ה-RAM. נשתמש ב-EPROM עם תוכנית המתחילה בכתובת 000 שלו, כלומר בכתובת 4000 מבחינת המערכת לאחר חיבורו.

1) בנה את מעגל הפענוח הבא



ציור 2-14: מעגלי פענוח ל-8085 ול-6800 לחיבור ROM.

2) חבר את ערוץ Y1 לקו ה-CS. הרץ את התוכנית שבסעיף 1.8. הסבר מדוע אינך מקבל דפקים בקו ה-CS.

3) שנה את התוכנית כך שתקבל דפקי CS במוצא מעגל הפענוח. רשום את התוכנית שכתבת.

4) שרטט את מעגל החיבור של ה-EPROM. היעזר בדפי הנתונים של הרכיב.

5) בנה את המעגל.

6) פנה לתא 4000 ובדוק אם תוכן התא זהה לתוכן שצריך להיות בו.

7) אם התוכן זהה, הרץ את התוכנית המתחילה בכתובת 4000.

8) אם התוכן אינו זהה, חזור על סעיף 2.9 (6 - 6).
שנה רק את התוכנית המורצת, לקריאה מהרכיב ולא לכתיבה בו.

2.11 חיבור RAM ו- ROM בעזרת מפענח

מטרת הניסוי היא לחבר יחידות זכרון RAM ו-EPROM לתחומי כתובות שונים בעזרת מפענח.

1) שרטט מעגל חשמלי, הקובע את ה-EPROM לתחום הכתובות 4000-47FF ואת ה-RAM לתחום הכתובות 5000-57FF, בעזרת מפענח. היעזר במעגל יצירת CS בעזרת מפענחים שמתואר בסעיף 2.4 ובדפי הנתונים של הרכיבים. כמפענח ניתן להשתמש ברכיב 74LS42.

2) בנה את המעגל על מטריצת החיבורים.

3) חזור על התרגילים שבסעיף 2.9.

4) חזור על התרגילים 6) עד 8) שבסעיף 2.10.

פרק 3 -פורטים

3.1 נושאי הפרק

א. בנית פורטים פשוטים - IN ו-OUT.
ב. הכרת מעגל הפענוח של ערכת הניסויים.
ג. הכרת פורטי הערכה.
ד. הכרת פורט הניתן לתיכנות.

רקע עיוני

3.2 הסבר כללי

במושג פורט מתכוונים ליחידה המשמשת את ה-CPU להתחברות אל העולם החיצון למערכת המיקרו-מחשב. PORT-IN קולט מספר בינארי מהעולם החיצון ומעבירו ל-CPU. PORT OUT קולט מספר בינארי מה-CPU ומעבירו לעולם החיצון. בדרך כלל, מכיל PORT OUT תאי LATGH, השומרים את המספר הבינארי ביציאותיו, עד לפניה הבאה של ה-CPU אליו. הפורטים נקראים גם יחידות I/O (INPUT/OUTPUT).

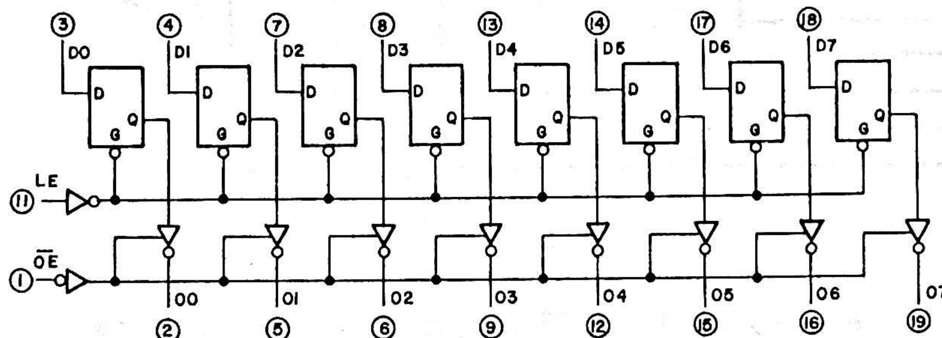
כל פורט תופס כתובת אחת במפת הזכרון. מבחינת ה-CPU הוא זהה לתא זכרון יחיד. ה-CPU אינו מבחין בהבדל. בדרך כלל, מיעדים תחום מסוים במפת הזכרון לשימוש יחידות ה-I/O.

ב-8085 קיימת אפשרות נוספת (מקובלת יותר) לבנית פורט, שבה בונים את הפורט מחוץ למפת הזכרון. ל-8085 יש שתי פקודות - OUT ו- IN שבזמן השימוש בהם עולה קו IO/M ל-HIGH, ובקוי הכתובת A0-A7 מופיעה הכתובת של הפורט המבוקש. מעגל הפענוח, במקרה זה, צריך לפענח את עלית קו IO/M ל-HIGH ומחצית מקוי הכתובת לשם יצירת CS לפורט.

קיימות יחידות הכוללות בתוכן מספר פורטים ולכן תופסות מספר כתובות.

3.3 פורט מוצא (PORT OUT) פשוט

כ-PORT OUT נשתמש ברכיב 74LS373. רכיב זה מורכב משמונה תאי LATCH בני סיבית אחת, בעלי כניסת ENABLE משותפת כמתואר בציור 3-1.



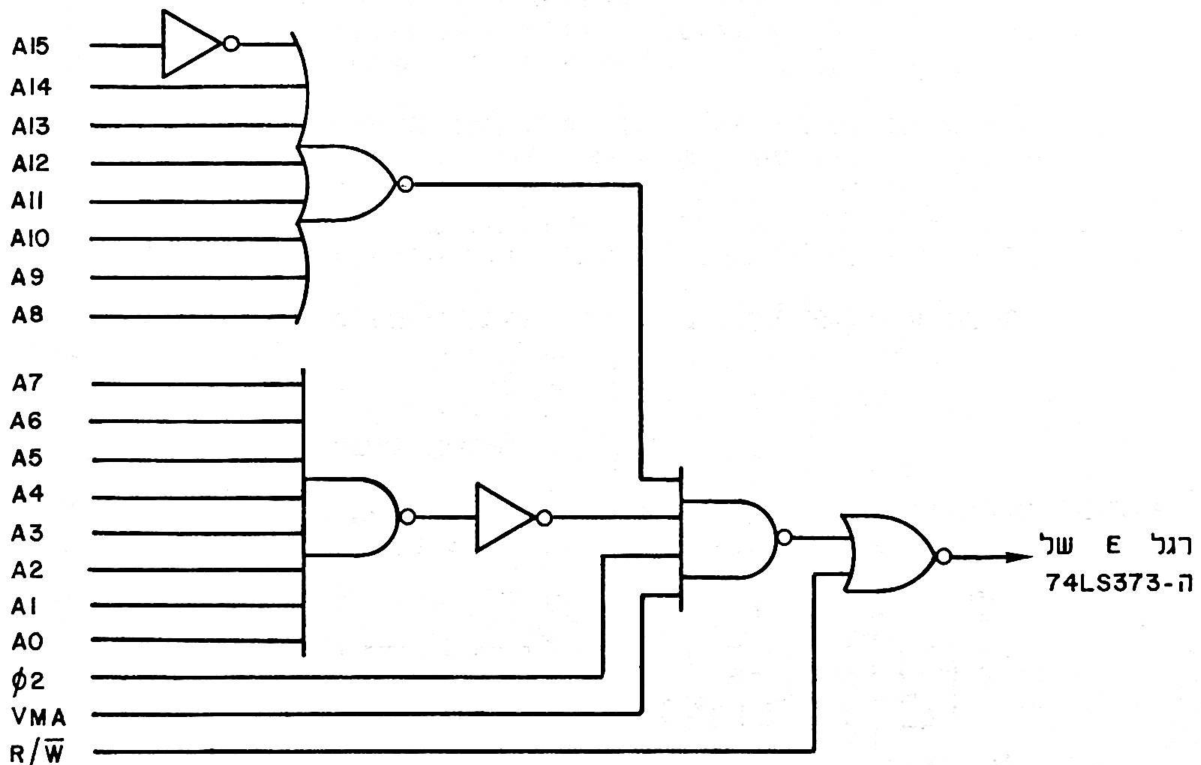
ציור 3 - 1 תיאור הרכיב 74LS373

ליציאות התאים יש חוצצים ולהם קו OE (OUTPUT ENABLE). אם רוצים שתוכו התאים יופיע באופן קבוע ביציאות הרכיב, נחבר את קו OE (רגל 1) באופן קבוע ל-GND. את קוי הנתונים (D0-D7) נחבר לקוים של מסלול הנתונים של ה-CPU.

יחידת פלט זו היא יחידה בת תא אחד בלבד ולכן תתפוס כתובת אחת. לשם דוגמא, נקבע פורט זה בכתובת 80FF, אם עובדים עם ה-6800. אם עובדים עם ה-8085, נקבע פורט זה בכתובת IO-FF. מעגל הפענוח צריך להעלות את קו E של ה-74LS373 (רגל 11) ל-HIGH, כאשר כותב ה-CPU בכתובת זו. במצב זה, יעבור המידע שעל קוי הנתונים לתאי הרכיב. עם סיום הכתיבה, ירד קו זה ל-LOW, והמידע שנמסר ישאר נעול בתאי ה-LATCH עד לכתיבה הבאה בכתובת זו.

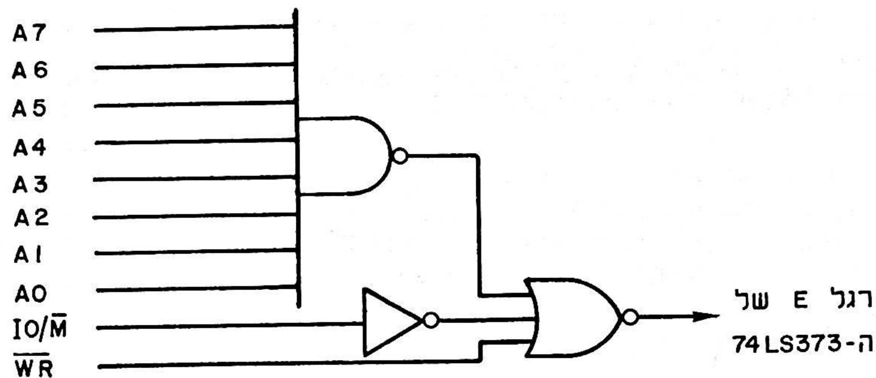
מעגל הפענוח חייב לפענח גם את קוי הבקרה.
ב-6800 קוים VMA, 02 ו-R/W.
ב-8085 קוים IO/M ו-WR.

מעגל הפענוח ל-6800 יראה כך:



ציור 2 - 3 מעגל פענוח ל-PORT OUT ל-6800.

מעגל הפענוח ל-8085 יראה כך:

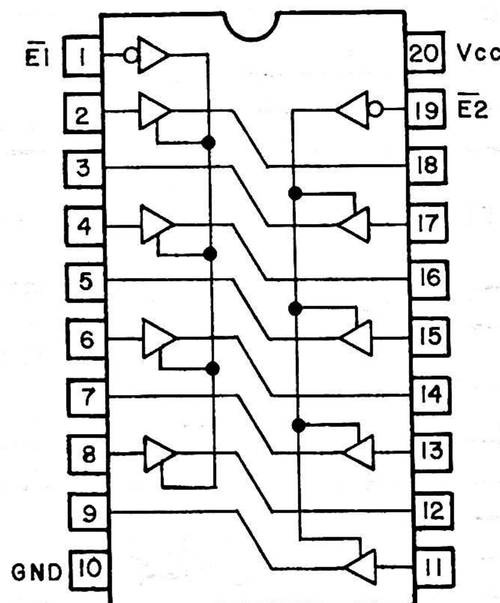


ציור 3 - 3 מעגל פענוח ל-PORT OUT ל-8085.

3.4 פורט מבוא (PORT IN) פשוט

כפורט מבוא פשוט נשתמש ברכיב 74LS244. זהו, למעשה, חוצץ בלבד. הוא אינו אוגר את נתוני הכניסה. הרכיב רק מפריד בין קוי הנתונים של מערכת המיקרו-מחשב לעולם החיצון. הוא נפתח רק כאשר ה-CPU פונה אליו לשם קריאת נתון ממערכת חיצונית המחוברת לכניסותיו.

הרכיב בנוי מ-8 חוצצים הערוכים בשתי רביעיות, כאשר לכל רביעיה קו ENABLE נפרד, כמתואר בציור 3-4. הורדת קו ה-ENABLE ל-LOW, מוציאה את החוצצים מ-3-STATE ומאפשרת מעבר המידע מקוי הכניסה לקוי היציאה של החוצצים.



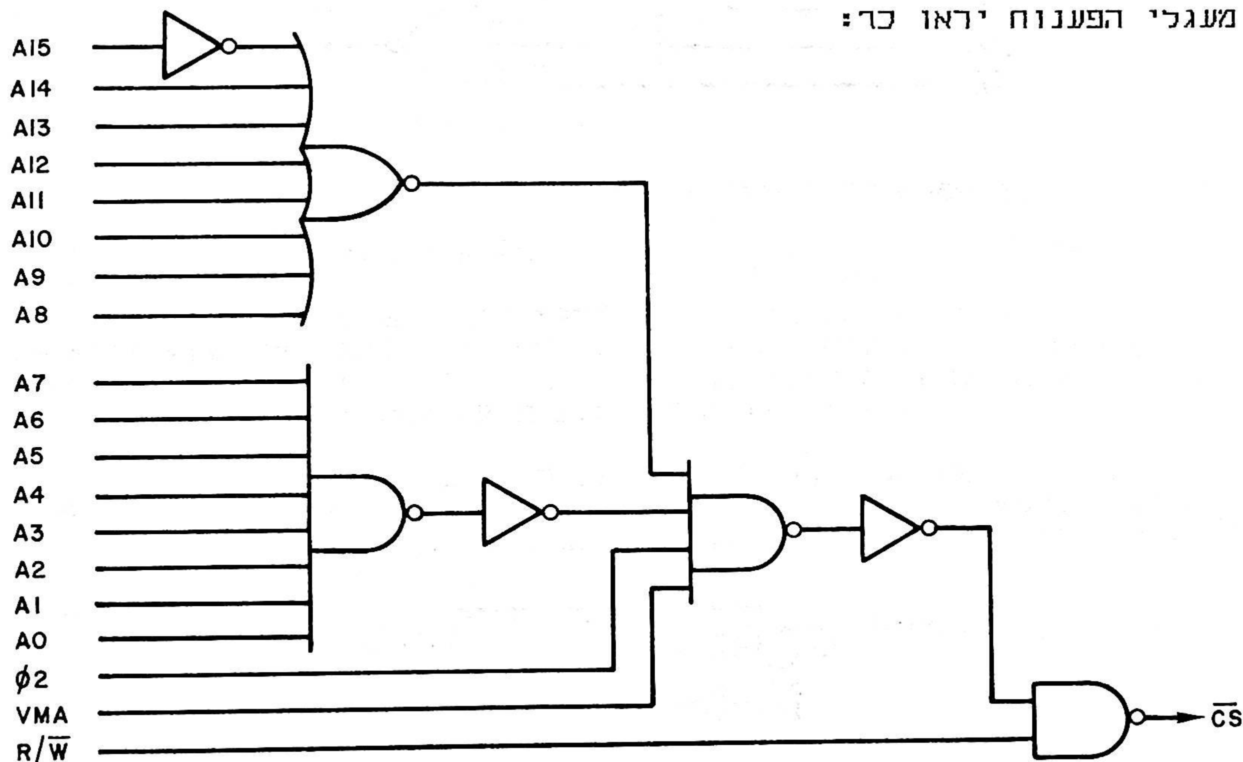
ציור 3 - 4 תיאור ה-74LS244

גם יחידת קלט זו תופסת כתובת של תא אחד בזכרון, בדוגמא, נקבע את כתובתה לכתובת של יחידת הפלט הקודמת (80FF ב-6800 או IO-FF ב-8085), בעזרת מעגל פענוח.

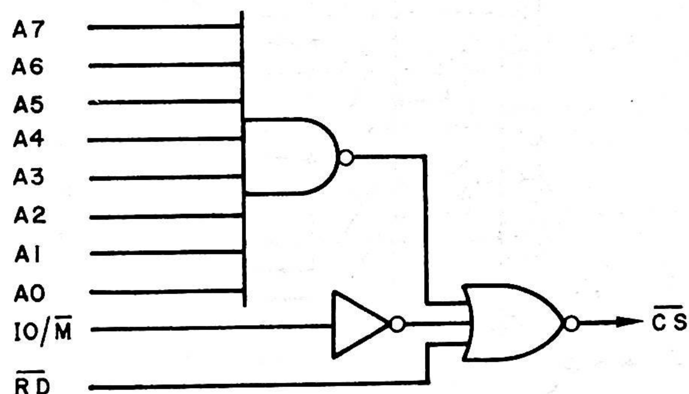
יציאת ה-CS של מעגל פענוח זה מתחבר לרגליים 1 ו-19 של ה-74LS244. קו זה צריך לרדת ל-LOW כאשר קורא ה-CPU מכתובת זו. מעגל הפענוח חייב לפענח גם את קוי הבקרה.

ב-6800 את הקוים VMA, 02 ו-R/W.
ב-8085 את הקוים IO/M ו-RD.

מעגלי הפענוח יראו כך:



ציור 5 - 3 מעגל פענוח ל-IN PORT ל-6800.

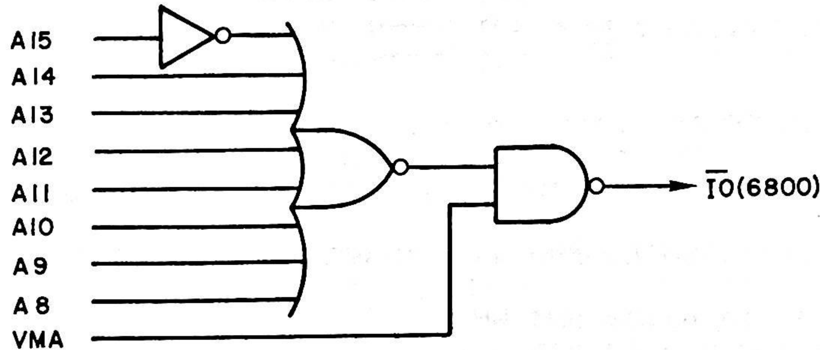


ציור 6 - 3 מעגל פענוח ל-IN PORT ל-8085.

3.5 מעגל הפענוח של ערכת הנסויים

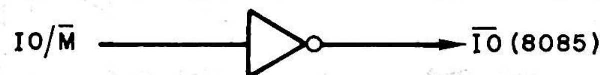
מעגל הפענוח של ערכת הניסויים מסוגל לפעול עם ה-6800 או עם ה-8085. את התאמתו ל-CPU המחובר קובעים בעזרת שלשה גשרים. מעגל הפענוח עצמו, הכולל 3 מפענחים (DECODERS), מפענח קוים מהצורה IO, RD ו-WR.

כדי לקבל ב-6800 קו IO', מפענחים את שמונה הסיביות העליונות של קוי הכתובת. בערכת הניסויים נקבע תחום הכתובות 8000-80FF כתחום ליחידות I/O. לכן יש ליצור קו היורד ל-LOW כאשר בקוים A8-A15 מופיעה כתובת אמיתית 80H (10000000). פענוח זה נעשה בעזרת המעגל הבא.



ציור 7 - 3 : מעגל ליצירת קו IO ל-6800

ב-8085 כדי לקבל קו IO' מספיק להעביר את קו IO/M' דרך מהפך.

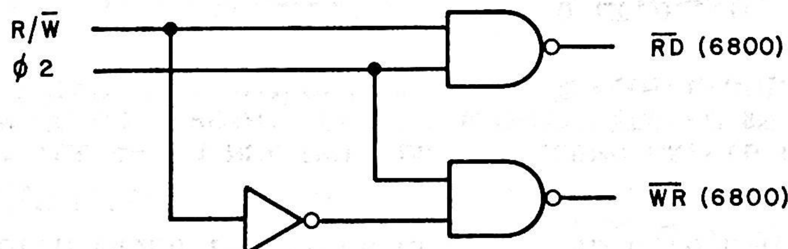


ציור 8 - 3 : מעגל ליצירת IO' ל-8085

יש מערכות המספקות ישירות קו IO' ולכן אין צורך להעבירו דרך מהפך.

הבחירה בין האפשרויות נעשית בעזרת גשר (JUMPER).

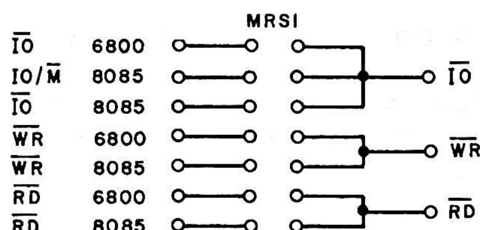
כדי לקבל קוים מהסוג RD' ו-WR' ב-6800, נעזרים במעגל המפענח את הקוים R/W' ו-O2 בצורה הבאה:



ציור 9 - 3 : מעגל ליצירת קוים RD' ו-WR' ל-6800

ב-8085 מסופקים קוים אלה על ידי ה-CPU. הבחירה בין קוי ה-6800 ל-8085 נעשית שוב בעזרת שני גשרים נוספים.

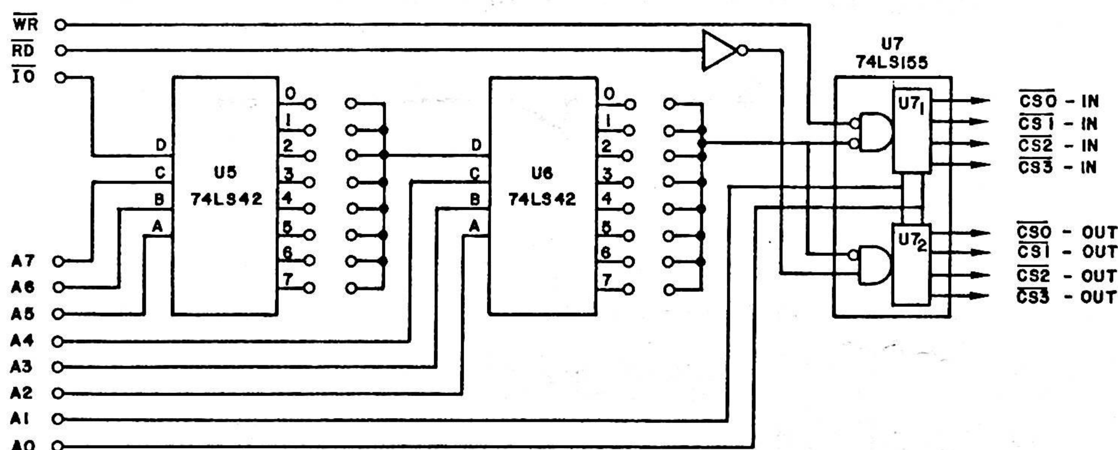
בסכומו של דבר נראית מערכת הגשרים בערכה בצורה הבאה:



ציור 3-10 : מערכת הגשרים בערכת הנסויים להתאמה לסוג ה-CPU.

קביעת מיקום שלשת הגשרים נעשה בהתאם לסוג המערכת המחוברת.

המשר המעגל מפענח את הקוים 'IO, 'RD, 'WR ואת שמונת קוי הכתובת הנמוכים (A0-A7) ויוצר בעזרתם 8 קוי CS בארבע כתובות I/O רצופות. קוים אלה נחלקים לארבעה קוי CS-IN וארבעה קוי CS-OUT. פעולת הפענוח נעשית בעזרת שלשה מפענחים, כמתואר בציור 3-11.



ציור 3-11: מעגל הפענוח של ערכת הניסויים

U5 מפענח את הקוים 'IO, A7, A6, ו-A5. כאשר 'IO ב-HIGH, נמצאים כל קוי המוצא של U5 ב-HIGH. כאשר 'IO ב-LOW (בפניה של ה-CPU ליחידת I/O), ירד ל-LOW אחד מקוי המוצא, בהתאם לצירוף הבינארי שבקוים A5-A7.

בחירת אחד הקוים בעזרת גשר גורמת לכך, שרק עם פנית ה-CPU ליחידת I/O בצירוף המתאים לקו זה ב-A5-A7 תגרום למיתוג U6.

לדוגמא, קביעת גשר על קו מוצא 2 של U5, תגרום שרק כאשר יפנה ה-CPU לכתובת XXXXX,IO-010 ימותג U6.

U6 מפענח את הקו הנבחר ב-U5 ואת הקוים A2-A4. כאשר הקו הנבחר ב-U5 ב-HIGH, נמצאים כל קוי המוצא של U5 ב-HIGH. כאשר הקו הנבחר ב-LOW, ירד ל-LOW אחד מקוי המוצא, בהתאם לצירוף הבינארי שבקוים A2-A4.

בחירת אחד הקוים בעזרת גשר, גורמת לכך שרק עם פנית ה-CPU ליחידת I/O בצירוף המתאים לקו הנבחר ב-U5 ובצירוף המתאים לקו הנבחר ב-U6, תגרום למיתוג U7.

אם נמשיך את הדוגמא הקודמת ונקבע גשר ביציאה 3 של U6, אזי רק כאשר יפנה ה-CPU לכתובת IO-010,011,XX ימותג U7.

U7 הוא מפענח המורכב למעשה משני מפענחים מסוג "1 מ-4", בעלי קוי כתובת משותפים. קוי הכתובת המתחברים ל-U7 הם הקוים A0,A1. לשני המפענחים יש 2 כניסות ENABLE לכל אחד, כמתואר בציור 3-11.

כאשר ממותג U7 על ידי U5 ו-U6, נעשית הבחירה בין U71 ל-U72 על ידי הקוים RD' ו-WR'. לעולם לא ימצאו שניהם ב-LOW. כאשר כותב ה-CPU בכתובת האמורה, יורד WR' ל-LOW וממותג U71. כאשר קורא ה-CPU מהכתובת האמורה, יורד RD' ל-LOW וממותג U72. הבחירה בין קוי ה-CS עצמם נעשית על ידי הקוים A0 ו-A1.

מעגל זה מאפשר חיבור שמונה פורטים נפרדים לתחום כתובות מוגדר. 4 פורטי מבוא ו-4 פורטי מוצא.

3.6 פורטי המערכת

במערכת קיימים 2 פורטי מבוא ו-2 פורטי מוצא לשם תרגול יישומים בעזרת פורטים. פורטים אלה משתמשים בקוים CS0 ו-CS1, כפי שמתואר בשרטוט החשמלי של המערכת. לרשות המשתמש עומדים 2 קוי CS-IN ו-2 קוי CS-OUT (CS2 ו-CS3). כדי להשתמש בקוים אלה, יש לקבוע גשרים ביציאות של U5 ו-U6 בהתאם לכתובת הנבחרת ולפנות בהתאם.

לדוגמא, אם הכתובת הנבחרת היא הכתובת מהדוגמא הקודמת (IO-010,011,XX) ומחברים פורט מבוא לקו CS2-IN, אזי כדי לקבל נתון מפורט זה יש לקרא אותו מהכתובת IO-010,011,10. אם רוצים לחבר פורט OUT לכתובת זו יש לחבר את קו ה-CS שלו לקו CS2-OUT. אם משתמשים בקו CS2, אז הכתובת שיש לפנות אליה היא הכתובת IO-010,011.11.

אחד מפורטי המוצא הבנויים בערכה הוא הרכיב 74LS373, והוא מבוקר על ידי קו CS0-OUT. הוא מכונה PORT OUT 1.

אחד מפורטי המבוא הבנויים בערכה הוא הרכיב 74LS244, והוא מבוקר על ידי הקו CS0-IN. הוא מכונה PORT IN 1.

שני הפורטים האחרים במערכת בנויים משני רכיבים 8212. אחד משמש כפורט מוצא, המנצל את הקו CS1-OUT. הוא מכונה PORT OUT 2. ה-8212 השני בנוי כפורט מבוא והוא מנצל את הקו CS1-IN. הוא מכונה PORT IN 2.

שני פורטים אלה יכולים לשמש כפורטים פשוטים מוצא ומבוא בהתאם. כדי להפעיל את ה-8212 כפורט מבוא פשוט יש לקצר את קו ה-STB לאדמה או לחברו לקו ה-CS של הרכיב. ב-8212 המחובר כפורט מוצא אין צורך לעשות דבר כדי להפעילו כפורט מוצא פשוט.

את המידע האגור ב-8212 ניתן לאפס בעזרת קו CLR.

ה-8212 מאפשר יצירת פסיקה ל-CPU והעברת נתונים באמצעות פסיקה. נושא זה יתואר בהמשך הפרקים.

3.7 פורט הניתן לתכנות

בשם פורט הניתן לתכנות (PROGRAMMABLE PORT) מתכוונים לרכיב, שבדרך כלל, יש לו מספר אופני פעולה ואופן הפעולה נקבע בתוכנה. גם קוויו ניתנים לקביעה בתוכנה כקוי OUTPUT או קוי INPUT או שלוב שלהם. רכיבים מהסוג הזה הם הרכיבים 8255, 6821 ועוד. רכיב כזה תופס יותר מכתובת אחת בזכרון. שני הרכיבים הנזכרים, תופסים ארבע כתובות כל אחד. לכן לרכיבים אלה מחברים בכניסת CS את מוצא מעגל פענוח של כל הקוים A3-A15 (במקרה של מיפוי זכרון) או A3-A7 במקרה של מיפוי I/O ב-8085. את הקוים A0,A1 מחברים ישירות לרכיב על מנת לאפשר בחירה בין הכתובות השונות.

ב-8255, נמצא באחת הכתובות (A0,A1=11) אוגר בקרה. פניה לאוגר זה והכנסת מספר מתאים קובעת את אופני פעולתו של הרכיב ויעוד פורטיו (IN או OUT). לרכיב שלשה פורטים - פורט A, פורט B ופורט C. סה"כ 24 קוי I/O.

ה-6821 בנוי משני חלקים סימטריים. כל חלק כולל אוגר בקרה, אוגר הקובע כיוון קוי הפורטים (IN או OUT) (DDR-DATA DIRECTION REGISTER) ואוגר נתונים. כל חלק כזה תופס שתי כתובות בזכרון למרות שהוא מורכב משלשה חלקים. כל אוגר בקרה בכל חלק תופס כתובת אחת בזכרון. הכנסת מספר לאוגר הבקרה, קובעת את אופי פעולתו של הפורט הקשור אליו וכן את יעוד הכתובת השניה שתופס אותו חלק - לאוגר כיוון קוי הנתונים (D.D.R) או לאוגר הנתונים.

בפניה לכתובת השניה שתופס אותו חלק, יגיע הנתון לאוגר הנתונים או לאוגר הכיוון בהתאם (או יתקבל הנתון במידה וקוראים נתון מהרכיב).

הרכיבים 8255 ו-6821 הם רכיבים מורכבים מאד ומיועדים למיגוון רחב של יישומים. אי אפשר לתאר את כל אפשרויות הפעלתם בחוברת זו ולכן נמנע מכך. על הקורא לדעת לקרוא את דפי הנתונים שלהם ולפעול

3.8 שאלות הכנה

1) תכנן מעגל פענוח לפורט מוצא לכתובת 8050 (ב-6800) או לכתובת IO-50 (ב-8085) חבר למעגל פענוח זה את הרכיב 74LS373.

2) תכנן מעגל פענוח לפורט מבוא לכתובת 8050 (ב-6800) או לכתובת IO-50 (ב-8085). חבר למעגל פענוח זה את הרכיב 74LS244.

3) כתוב תוכנית המעבירה נתון מפורט מבוא לפורט מוצא בלולאה אינסופית. הפורטים הם אלה שתוארו בסעיפים 1) ו-2).

4) תאר מעגל מלא לחיבור הרכיב 8255 בתחום הכתובות IO-04 (8004 ב-6800) עד IO-07.
כתוב תוכנית הקובעת את פורט A כפורט מבוא ואת פורט B ופורט C כפורטי מוצא. התוכנית גם מעבירה נתון מפורט A לפורט B בלולאה אינסופית. בצע זאת גם אם ברשותך מערכת עם המיקרופרוססור 6800.

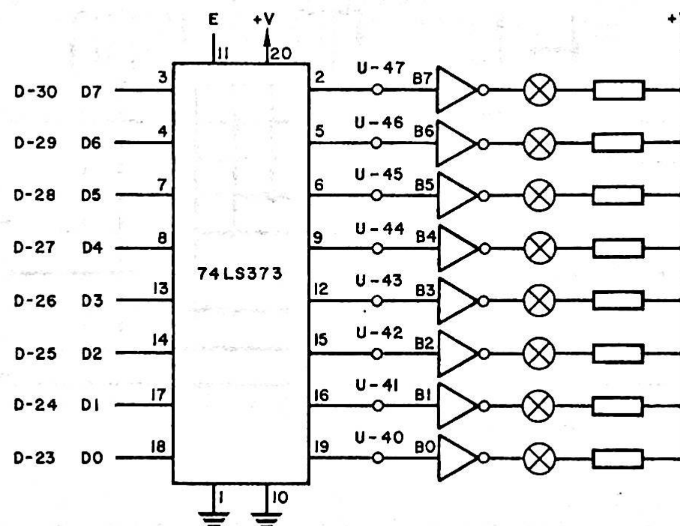
5) תאר מעגל מלא לחיבור הרכיב 6821 בתחום הכתובות המתואר בסעיף הקודם.
כתוב תוכנית הקובעת את פורט A כפורט מבוא ואת פורט B כפורט מוצא. התוכנית גם מעבירה נתון מפורט A לפורט B בלולאה אינסופית.
בצע זאת גם אם ברשותך מערכת עם המיקרופרוססור 8085.

מהלך הניסוי

3.9 בנית פורט מוצא פשוט

מטרת הניסוי היא לבנות פורט מוצא פשוט בעזרת הרכיב 74LS373 ולחבר את יציאותיו ל-"לדים" של ערכת התרגול.

1) בנה את מעגל הפענוח המתואר בציור 3-1 על מטריצת החיבורים, בהתאם לסוג המיקרופרוססור שברשותך.



ציור 3-12: חיבור פורט מוצא ל "לדים" של ערכת הנסויים.

2) הוסף למעגל הפענוח את יחידת ה-74LS373 כמתואר בציור 3-12.

החיבור לדוחפים ול"לדים" נעשה דרך נקודות החיבור במטריצה העליונה

3) כתוב תוכנית המעבירה לכתובת הפורט את המספר 0F. הרץ את התוכנית ובדוק אם ארבע הנוריות המתאימות אכן נדלקות. אם לא נדלקו, דלג לסעיף 5 לשם איתור התקלה.

4) העבר מספרים שונים אל הפלט ובדוק אם ערכם הבינארי אכן מופיע ב"לדים".

5) הרץ את התוכנית בלולאה אינסופית ובדוק את האותות המופיעים ביציאות השערים. בדוק במיוחד את קו ה-CS.

6) שרטט מעגל מלא הכולל CPU, מעגל פענוח ומעגל יחידת הפורט שבנית.

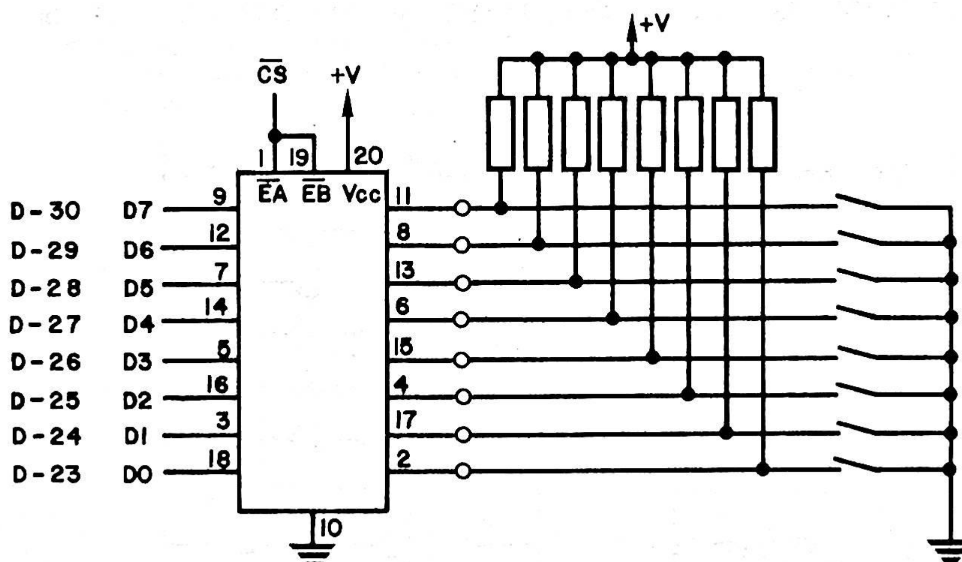
3.10 בנית פורט מבוא פשוט

מטרת הניסוי היא לבנות פורט מבוא פשוט בעזרת הרכיב 74LS244 ולחבר את כניסותיו למפסקי ערכת הנסויים.

1) בנה את מעגל הפענוח המתואר בציור 3-2 על מטריצת החיבורים, בהתאם לסוג המיקרופרוססור שברשותך. שים לב שאינך צריך לשנות הרבה במעגל הקודם, לכן בנה את פורט המבוא ליד פורט המוצא והיעזר במעגל הפענוח הקודם.

2) הוסף למעגל הפענוח את יחידת ה-74LS244 כמתואר בציור 3-13.

החיבור למפסקי הערכה יתבצע דרך נקודות החיבור שלידם.



ציור 3-13: חיבור פורט מבוא למפסקי הערכה.

3) כתוב תוכנית הטוענת את הצובר מפורט המבוא ולאחר מכן מוציאה תוכן הצובר לפורט המוצא. שים לב ששני הפורטים הם בעלי אותה הכתובת IO-FF.
קבע מספר בינארי כלשהו במפסקים, הרץ את התוכנית ובדוק אם המספר הופיע בנוריות הפלט. אם התוכנית אינה מתבצעת עבור לסעיף 4).

4) שנה את התוכנית לריצה בלולאה אינסופית. הרץ אותה. בדוק את האותות המופיעים ביציאות השערים. שינוי במפסקי הקלט צריך לגרום לאותו שינוי בנוריות הפלט.

5) שרטט מעגל מלא הכולל את פורט המבוא, פורט המוצא, מעגל הפענוח וה-CPU.

3.11 שימוש במעגל הפענוח של הערכה

מטרת הניסוי היא לתרגל שימוש במעגל הפענוח של ערכת הניסויים ולחבר אליו את פורט המבוא ופורט המוצא שנבנו בניסויים 3.9 ו-3.10.

1) נתק את מעגלי הפענוח של התרגול הקודם. השאר את ה-74LS373 וה-74LS244 מחוברים ל"לדים", למפסקים ולקוי הנתונים.

2) בדוק שמערכת הגשרים MRS1 מחוברת בהתאם לסוג ה-CPU שברשותך.

3) קבע 2 גשרים ב-MRS2 וב-MRS3 לכתובת פורט FF.

4) חבר את קו CS3-OUT לכניסות ה-CS של ה-74LS244.

5) חבר את קו CS3-OUT דרך מהפך לכניסת ה-ENABLE של ה-74LS373.

6) חזור על תת-הסעיפים 3) ו-4) של סעיף 3.10. התוצאה צריכה להיות זהה.

7) קבע את כתובת פורט המבוא לכתובת IO-06 ואת כתובת פורט המוצא לכתובת IO-07 (על ידי שנוי הגשרים ב-MRS2 וב-MRS3 בלבד).

8) חזור על סעיף 6) עם תוכנית אחרת.

3.12 שימוש בפורטי הערכה

מטרת הניסוי היא לתרגל שימוש בפורט המוצא ובפורט המבוא של ערכת הניסויים על ידי חיבורם אל ה"לדים" של הערכה ואל מפסקי הערכה בהתאמה.

1) השאר את שני הגשרים מהניסוי הקודם ב-MRS2 וב-MRS3.

2) נתק את הרכיבים ממטריצת החיבורים.

3) חבר את יציאות PORT OUT-1 לכניסות הדוחפים ל"לדים".

4) חבר את כניסות PORT IN-1 לחיבורי המפסקים.

5) מערכת הגשרים שקבעת בתת-סעיף 7) שבניסוי 3.11 קובעת את IN-1 PORT ואת PORT OUT-1 לכתובת IO-04. בדוק זאת.

6) שנה את התוכנית מהניסוי הקודם כך שתעביר את תוכן המפסקים אל הנוריות בלולאה אינסופית דרך פורטי ערכת התרגול IN-1 ו-OUT-1.

3.13 חיבור פורט הניתן לתכנות (PROGRAMMABLE PORT)

מטרת הניסוי היא לתרגל חיבור פורט הניתן לתכנות אל ערכת הניסויים והפעלתו, תוך כדי עזר בקו CS של מעגל הפענוח שבערכה.

1) היעזר ביציאות CS של מעגל הפענוח בערכת הניסויים וחבר על המטריצה פורט הניתן לתיכנות מסוג 6821 או 8255. היעזר בדפי הנתונים של הרכיב.

2) חבר את יציאות פורט A של הרכיב לכניסות הדוחפים של ה"לדים". פורט A ישמש כפורט מוצא.

3) חבר את יציאות פורט B של הרכיב לחיבורי המפסקים. פורט B ישמש כפורט מבוא.

4) בדוק לפי מצב הגשרים ב-MRS2 וב-MRS3 את הכתובות שתופס הרכיב וכתובות פורטיו.

5) כתוב תוכנית המבצעת את השלבים הבאים:
   א) קובעת את פורט A כפורט מוצא.
   ב) קובעת את פורט B כפורט מבוא.
   ג) מעבירה את תוכן פורט B אל הצובר (טוענת את הצובר במצב המפסקים).
   ד) מעבירה את תוכן הצובר אל פורט A (אל הנוריות).
   ה) חוזרת לסעיף ג).

6) הרץ את התוכנית.

7) שנה את מצב המפסקים ובדוק את השפעת השנויים על נוריות הפלט.

8) שרטט מעגל של חיבורי הרכיב אל הערכה.

פרק 4 - תצוגה

4.1 נושאי הפרק

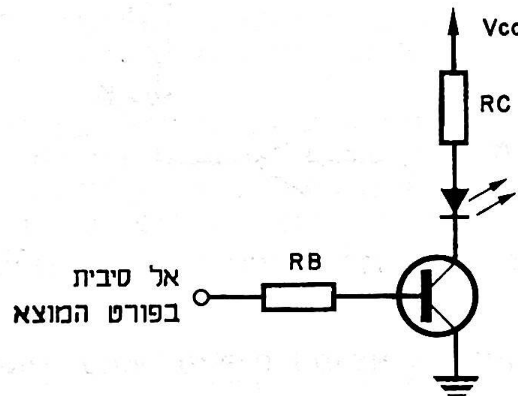
א. LEDS ואופן חיבורם לפורט מוצא.
ב. תצוגת .7-SEG ואופן הפעלתה בריבוב (MULTIPLEXING).
ג. המרה מ- .B.C.D ל- .7-SEG ושימוש בטבלאות.
ד. תצוגת מטריצת נקודות.

רקע עיוני

4.2 LEDS ואופן חיבורם לפורט מוצא

ה-LED (LIGHT EMITTING DIODE) הוא דיודה הפולטת אור כאשר זורם דרכה זרם. עוצמת האור תלויה בעוצמת הזרם דרכה. כדי לקבל עוצמת אור סבירה, מספיק להעביר דרכה זרם בין 5MA ל-10MA.

את ה"לד" ניתן לחבר ישירות אל פורט המוצא, אבל רצוי יותר לחברו אל מעגל דוחף שיופעל על ידי הפורט. מעגל הדוחף, הנקרא DRIVER, יכול להיות בנוי ממעגל טרנזיסטורי פשוט.



ציור 1 - 4 מעגל דוחף טרנזיסטורי ל"לד".

RC ו-RB נקבעים בהתאם לזרם שרוצים שיעבור ב"לד" (IC). נניח שרוצים לקבל IC=10MA. מפל המתח בהולכה על הדיודה ועל VCE (ברוויה) הוא 1.5V בערך. מכאן שעל RC יפול מתח של 3.5V. מכאן

$$RC = \frac{VRC}{IC} = 350\ OHM$$

מקובל לחבר נגד של 330 OHM כאשר רוצים לקבל זרם בסדר גודל של 10MA ו-680 OHM כאשר רוצים לקבל זרם של 5MA.

הטרנזיסטור ממותג כאשר סיבית המוצא של הפורט, המחוברת אליו, נמצאת ב-HIGH (VOH). VOH - מתח המוצא ה"גבוה" המקובל ברכיבים המסתיימים ב"עמוד טוטם" הוא 3.7V בערך. IB הנדרש למיתוג הטרנזיסטור יחשב לפי B MIN.

$$IB = \frac{IC}{B\ MIN}$$

ולכן

$$RB = \frac{VOH - VBE(SAT)}{IB}$$

RB מתקבל בסדר גודל של 5K OHM בערך, כאשר הקטנתו מעבר לכך גורמת להכנסת הטרנזיסטור לרוויה עמוקה. בתצוגה, בעיית זמני המיתוג לא קריטית בדרך כלל ולכן אין זה נורא.

ניתן גם למתג את ה"לד" בעזרת מעגל דוחף משולב, כמו ה-7406 או MC1416 בצורה הבאה.



ציור 2 - 4 מעגל דוחף משולב ל"לד".

גם כאן ה"לד" מאיר כאשר סיבית המוצא של הפורט נמצאת ב-HIGH.

4.3 בבואה לפורט מוצא

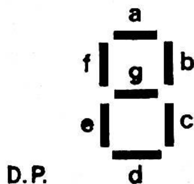
ברוב פורטי המוצא, אין אפשרות לקרוא את תוכנם. לדוגמא, נניח שרוצים לבצע "אור רץ" על שורת ה"לדים" המחוברת לפורט מוצא. השיטה היא לבצע זאת בלולאה. מתחילים בהדלקת הנורית הימנית ומבצעים השהיה.

אם הפורט היה מתנהג כמו צובר, היינו פשוט מסובבים שמאלה וחוזרים לביצוע השהיה. אם הפורט היה מתנהג כמו תא בזכרון, היינו מעבירים את תוכנו לצובר, מסובבים שמאלה ומחזירים לתא. ב-6800 ניתן לבצע פקודת "סובב" (ROTATE) ישירות על תא בזכרון.

מכיוון שאי אפשר לקרוא תוכו של פורט מוצא ולהעבירו לצובר, אין ברירה אלא לקבוע תא זכרון כבבואה של פורט המוצא. כל פעם שמעבירים נתון לפורט המוצא, מעבירים אותו נתון גם לתא זה. הקריאה של הנתון תיעשה מהתא.

4.4 תצוגת 7-SEG. והפעלתה בצורה דינמית על ידי ריבוב (MULTIPLEXING).

תצוגת 7-SEG. מורכבת כידוע משבעה "לדים" בצורת הספרה 8, ו"לד" שמיני כנקודה עשרונית.



ציור 3 - 4 ארגון ה-SEGME NTS ביחידת 7-SEG.

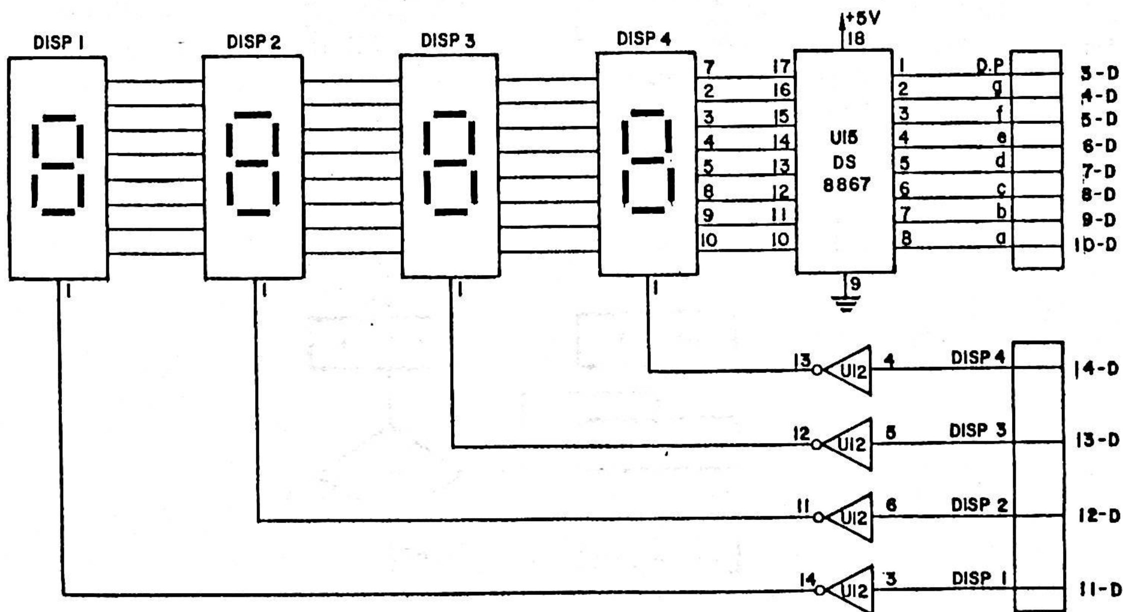
את שמונת ה"לדים" ניתן לחבר דרך מעגל דחיפה אל סיביות פורט מוצא. העלאת סיבית מוצא ל-HIGH גורמת להדלקת אחד ה"לדים".

כאשר רוצים להפעיל תצוגה עם מספר יחידות 7-SEG., ניתן לחבר כל יחידה לפורט מוצא אחר. לשיטה זו שני חסרונות כבדים:

א) שימוש במספר רב של פורטי מוצא.

ב) צריכת זרם גדולה. בשיטה זו כל יחידת תצוגה מופעלת כל הזמן. הזרם הנצרך הוא סכום הזרמים שצורכות כל היחידות.

השיטה המקובלת להפעלת מספר יחידות תצוגה, היא הפעלתן בצורה דינמית. בשיטה זו מחברים את שמונה הלדים של כל יחידות התצוגה במקביל, דרך מעגל דחיפה, אל יציאותיו של פורט מוצא אחד. נכנה פורט זה בשם פורט A. מספר בינארי המופיע ביציאותיו של פורט זה, מופיע למעשה בכניסות של כל יחידות ה- 7-SEG.. את כניסות המתח של יחידות התצוגה מחברים, דרך מעגל דחיפה, לפורט נוסף - פורט B. ראה ציור.



ציור 4 - 4 מעגל דחיפה ליחידות 7-SEG.

השלבים בהפעלת תצוגה בשיטה זו הם:

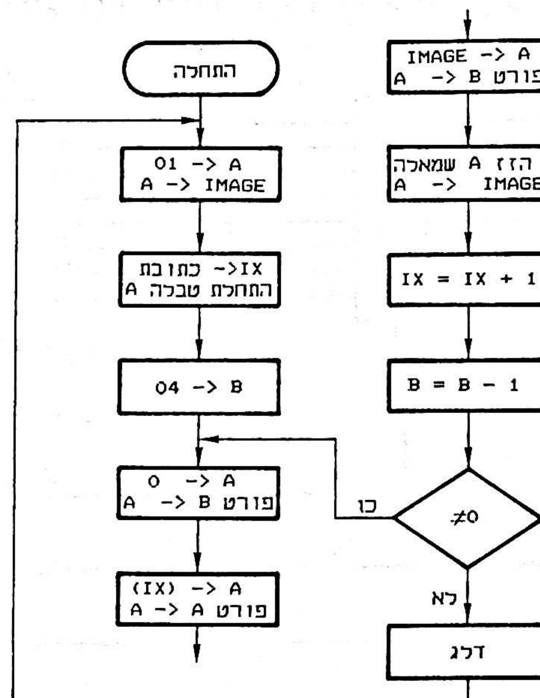
1) הוצאת המספר מיועד ליחידת התצוגה הראשונה בעזרת פורט A.
2) חיבור מתח ליחידה זו בעזרת פורט B.
3) השהיה קצרה (ניתן לוותר עליה).
4) ניתוק המתח מיחידות התצוגה (00 לפורט B).
5) הוצאת המספר המיועד ליחידת התצוגה השניה בעזרת פורט A.
6) חיבור מתח ליחידה זו בעזרת פורט B.
7) השהיה קצרה.
8) ניתוק המתח מהיחידות
וכן הלאה. לאחר הפעלת יחידת התצוגה האחרונה חוזרים ליחידת התצוגה הראשונה וחוזר חלילה.

בשיטה זו, עסוק ה-CPU כל הזמן בהפעלת התצוגה. ניתן לבנות תוכנית זו בצורה של תוכנית משנה ולקרוא לה כל פעם לשם הצגה פעם אחת ביחידת התצוגה.

לפניך תרשים זרימה לתוכנית, המציגה תוכן של ארבעה תאים בזכרון, בארבע יחידות תצוגה. פורט A מוציא את המספר לתצוגה. פורט B מפעיל את יחידת התצוגה הדרושה. אוגר B משמש כמונה. בתוכנית זו, המפעילה 4 יחידות תצוגה, הוא סופר עד 4.

התוכנית משתמשת בתאי זכרון מה-RAM. כותב התוכנית בוחר את התאים כרצונו. תא אחד ישמש כבבואה לפורט B ונכנה אותו בשם IMAGE. ארבעה תאים רצופים אחרים, ישמשו כטבלה לאיחסון המספרים המוצגים ביחידות התצוגה (טבלה A). IX מציין INDEX REGISTER (אוגר מציין) או את H,L ב-8085.

תרשים זרימה א'



במקרה של שימוש בתוכנית כתוכנית משנה, מוותרים על ה- דלג בסוף ורושמים במקומו "חזור מתוכנית משנה".

4.5 המרה מ- .B.C.D ל- .7-SEG וחיפוש בטבלאות

המספרים הבינאריים הנמצאים בטבלה A ומיועדים לתצוגה, מפעילים את ה"לדים" שבתצוגה. כל סיבית מפעילה SEGMENT אחר. נקרא לצופן זה צופן .7-SEG. כאשר כותבים תוכנית המיועדת להוציא מספרים לתצוגה עשרונית, מוציאים מספרים בצופן .B.C.D. נדרשת, אם כן, תוכנית שממירה טבלה של מספרים בצופן .B.C.D(טבלה B), לטבלה של מספרים בצופן .7-SEG (טבלה A של התוכנית הקודמת).

כדי לבצע זאת נעזרים בטבלת חיפוש (טבלה זו נקראת LOOK UP TABLE - LUT) - טבלה C. טבלה זו מכילה את המספרים הבינאריים היוצרים את הספרות השונות ביחידות התצוגה. לדוגמא, בתא הראשון יהיה המספר הבינארי היוצר את ספרה O - . בתא השני את הספרה - : וכו הלאה.

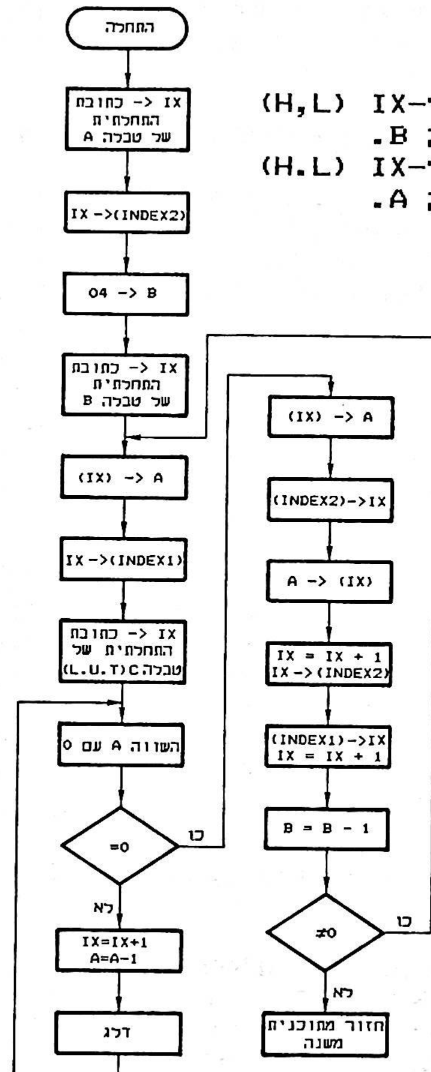
התוכנית פונה לתא הראשון בטבלה B (טבלת המספרים להמרה) ומעבירה את תוכנו לצובר. לאחר מכן, עוברת לתא שמספרו הסדורי בטבלה C שווה ל- A. לדוגמא, אם A שווה ל-3, התוכנית פונה לתא שמספרו הסדורי 3 בטבלה C (זהו התא הרביעי כי הראשון מספרו הסידורי הינו אפס). תא זה יכיל את המספר הבינארי היוצר את הספרה 3 בצופן .7-SEG. את תוכו תא זה מעבירה התוכנית לצובר ומשם לתא הראשון בטבלה A. לאחר מכן פונה התוכנית לתא השני בטבלה B. כך הלאה עד לסיום תאי טבלה B.

התוכנית בנויה כתוכנית משנה, מפני שקוראים לה פעם אחת, בכל פעם שנדרשת המרה.

כאמור, משתמשת התוכנית בשלוש טבלאות. כותב התוכנית חייב לקבוע את הכתובת ההתחלתית שלהן ואת אורכן. אורך טבלאות A ו-B הוא כמספר יחידות התצוגה. טבלה C (LOOK UP TABLE) מכילה עשרה מספרים. את טבלה C יש למלא יחד עם כתיבת התוכנית. את טבלה B יש למלא לפני הקריאה לתוכנית. טבלה A היא הפלט של התוכנית. טבלה A צריכה להיקבע בתחום כתובות מתאים, כך שהתוכנית של תרשים זרימה א' תציג תוכן טבלה זו בתצוגה.

תרשים זרימה ב'

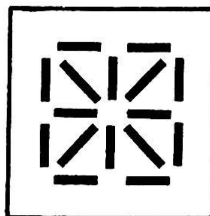
אוגר B - משמש כמונה.
INDEX1 - כתובת איחסון ל-IX (H,L) המציין של טבלה B.
INDEX2 - כתובת איחסון ל-IX (H.L) המציין של טבלה A.



4.6 תצוגות אחרות ושיטות מיתוג שונות

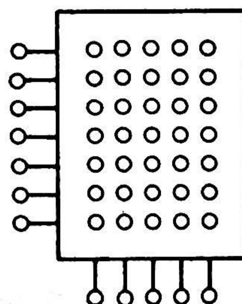
קיימות תצוגות נוספות כמו התצוגות האלפא נומריות. לא נרחיב עליהן את הדיבור, אלא נזכיר חלק מהן בקצרה.

א) תצוגת 16-SEG.
בנויה כמתואר בציור 4-5. מיתוגה דומה למיתוג יחידת 7-SEG.



ציור 4 - 5 תצוגת 16-SEG.

ב) תצוגת מטריצת נקודות.
תצוגה זו בנויה במערך של נקודות. קיימים מספר סוגים של מערכים 5X7, 7X9, 9X11. לדוגמא, מערך 5X7.



ציור 6 - 4 תצוגת מטריצת נקודות 5X7

מיתוג יחידה כזו מחייב שימוש במעגל מיתוג דינמי של כל עמודה בנפרד. מעגל כזה בנוי בחומרה ונעזר ב-ROM.

קיימות שיטות שונות להפעלת תצוגה. השיטה שתוארה בסעיפים הקודמים היא שיטה המשתמשת בתוכנה בלבד. קיימות שיטות המשלבות תוכנה עם חומרה או פועלות בחומרה בלבד. אחת השיטות נפוצות להפעלת תצוגה בחומרה משתמשת ברכיב 8279. מכיוון שרכיב זה מטפל גם בלוח מקשים נתאר את השימוש בו בפרק הבא.

4.7 שאלות הכנה

1) רשום תרשים זרימה ותוכנית ליצירת "אור רץ" בפורט IO-05.
האור "רץ" מימין לשמאל.

2) המר את תרשים זרימה א' לתוכנית כאשר
פורט A הוא בכתובת IO-04
ופורט B הוא בכתובת IO-05.
את כתובת טבלת המספרים המוצגים קבע כרצונך בהתאם ל-RAM במערכת המיקרו-מחשב שברשותך.

3) המר את תרשים זרימה ב' לתוכנית כאשר את כתובת טבלת המספרים להמרה קבע כרצונך בהתאם ל-RAM במערכת המיקרו-מחשב שברשותך.
את המספרים המומרים העבר לטבלה שהשתמשת בה בתוכנית שבסעיף 2).
קבע גם את כתובת טבלת החיפוש. את טבלת החיפוש תמלא רק לאחר ביצוע ניסוי 4.9.

4) שרטט מעגל מיתוג של ארבע יחידות תצוגה מסוג מטריצת נקודות 5X7.

4.8 8212 כפורט מוצא פשוט

מטרת הניסוי להפעיל את שמונת ה"לדים" בעזרת ה-8212 המחובר כפורט מוצא. לשם שימוש ב-8212 כפורט מוצא פשוט אין צורך לעשות דבר, מלבד להתחבר ליציאותיו. קוים CLR ו-STB מחוברים עם נגדי משיכה ל-+V.

1) חבר את יציאות PORT OUT-2 לכניסות הדוחפים של ה"לדים".

2) חבר גשרים לקביעת הפורט בכתובת IO-05.

3) שרטט מעגל חשמלי הכולל את ה-8212, הדוחפים, ה"לדים" והנגדים המחוברים אליהם

4) רשום תוכנית ליצירת "אור רץ" בנוריות הפלט והרץ אותה.

4.9 הפעלת יחידת .7-SEG אחת

מטרת הניסוי היא להפעיל יחידת .7-SEG אחת באמצעות פורט מוצא, ולזהות איזה סיבית מוצא מפעילה כל SEGMENT.

1) חבר את יציאות PORT OUT-1 לכניסות המקביליות של יחידות ה-.7-SEG.

2) תקע את יחידת התצוגה ללוח התקיעה. חבר את כניסת המתח של יחידת התצוגה הראשונה ל-+V.

3) מערכת הגשרים של הניסוי הקודם קובעת את PORT OUT-1 בכתובת IO-04. הוצא לפורט זה את המספרים הבינאריים המפעילים כל אחד SEGMENT אחר, בזה אחר זה. המספרים הם 01, 02, 04, 08, 10, 20, 40 ו-80.

4) רשום ליד כל SEGMENT את הסיבית המפעילה אותו. למשל
A - D0
B - D1

וכו'.

5) הכן לעצמך את טבלה C (טבלת החיפוש - .L.U.T). בטבלה רשום לפי הסדר את המספרים הבינאריים שיש להוציא ליחידות ה- .7-SEG כדי לקבל את הספרות השונות.

6) הוצא את המספרים האלה אל יחידת ה- .7-SEG המחוברת, ובדוק אם אכן קבלת את הספרות הדרושות.

4.10 הפעלת תצוגה דינמית

מטרת הניסוי להפעיל את ארבע יחידות ה- .7-SEG של המערכת בצורה דינמית על ידי ריבוב (MULTIPLEXING). לשם כך נעזרים בשני פורטי

מוצא ובתוכנית שנכתבה לתרשים זרימה א'. בשלב זה עדיין אין ממירים מ- B.C.D. ל- 7-SEG. המספרים הנשלחים לתצוגה (נמצאים בטבלה A), הם מספרים להפעלת תצוגת 7-SEG. שמצאנו בניסוי 4.9.

1) קבע כתובת לטבלה A. טבלה A היא בת ארבעה תאים.

2) קבע כתובת ל-IMAGE.

3) רשם את התוכנית לתרשים זרימה א' שבסעיף 4.4.

4) מלא את טבלה A בארבעה מספרים בינאריים שמצאת בסעיף 4.9.

5) הרץ את התוכנית ובדוק אם הספרות המתאימות אכן מופיעות ביחידות התצוגה.

4.11 הפעלת התצוגה כולל המרה מ-B.C.D ל- 7-SEG.

מטרת הניסוי להפעיל את ארבע יחידות התצוגה של המערכת כמו שנעשה בניסוי הקודם, יחד עם המרה מצופן B.C.D. לצופן 7-SEG.
המספרים המיועדים לתצוגה יהיו מספרים בצופן B.C.D. בטבלה B. הרצת התוכנית לתרשים זרימה ב' תמיר אותם לצופן 7-SEG., לטבלה A. הרצת התוכנית לתרשים זרימה א' תציג אותם ביחידות התצוגה.

1) קבע כתובת לטבלה B (ארבעה תאים). מלא את התאים בארבעה מספרים (לדוגמא - 04, 06, 07, 09). בדוק שטבלה B אינה חופפת לטבלה A או לתוכנית מהסעיף הקודם.

2) קבע כתובות ל-INDEX1 ול-INDEX2. זכור שכל INDEX תופס שני תאי זכרון.

3) קבע כתובת התחלתית לטבלה C (עשרה תאים). מלא אותה במספרים הבינאריים שמצאת בסעיף 4.9.

4) רשום תוכנית לתרשים זרימה ב' שבסעיף 4.5. רשום תוכנית זו כתוכנית משנה.

5) הוסף בתחילת התוכנית שבסעיף 4.10 קריאה לתוכנית המשנה. קריאה זו תגרום להמרת המספרים מצופן B.C.D שבטבלה B לצופן 7-SEG. בטבלה A. לאחר החזרה מתוכנית המשנה התוכנית תציג בלולאה אינסופית את המספרים שבטבלה B בתצוגה.

6) הרץ את התוכנית עם מספרים שונים בטבלה B ובדוק אם אכן היא פועלת.

פרק 5 - מפסקים ומקשים

5.1 נושאי הפרק

א. קריאת מצב מפסקים וזיהוי המפסק הנלחץ.
ב. סריקת מקשים.
ג. ביטול ריטוט (BOUNCING) בתוכנה ובחומרה.
ד. לוח מקשים וסריקה דינמית בתוכנה.
ה. הכרת ה-8279

רקע עיוני

5.2 קריאת מצב מפסקים וזיהוי המפסק הנלחץ

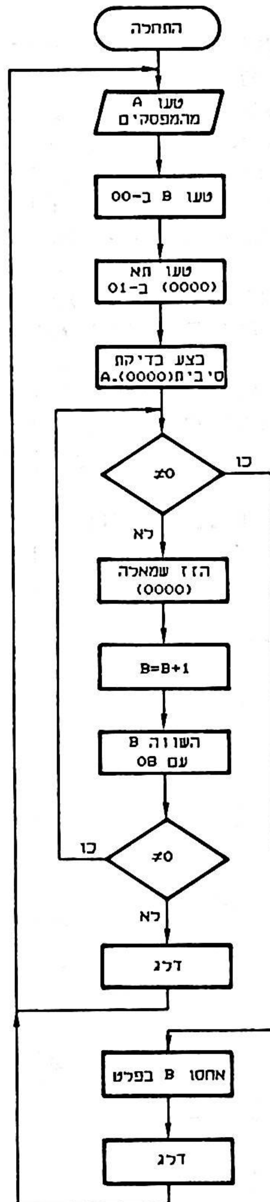
בפרק 3 ראינו כיצד לחבר מפסקים אל פורט מבוא, וכיצד לקרוא את מצב המפסקים ולהעבירו לפורט מוצא, המוציא אותו לנוריות. כדי להגיב להפעלת מפסק כלשהו, יש לזהות עליה ל-"1" בסיבית המקבילה לו, במספר הבינארי שהתקבל מהמפסקים. הבדיקה נעשית בעזרת הפעולה הלוגית AND.

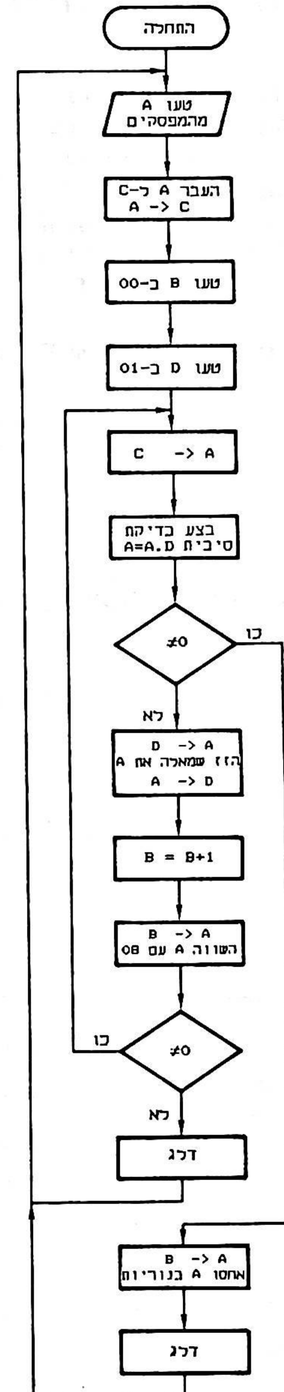
לדוגמא, נניח שרוצים לבדוק סיבית D1 (הסיבית השניה) והמספר נמצא בצובר. מבצעים פעולת AND על הצובר, במיעוו מיידי, עם המספר הבינארי 00000010. האפסים יאלצו את הסיביות המקבילות להם בצובר לרדת ל"0", לא משנה מה היה מצבו הקודם. רק סיבית D1 תישאר במצבה הקודם. אם לאחר הפעולה הצובר מכיל אפס (דגל האפס עלה ל-"1") אזי D1 היה "0". אם לאחר הפעולה בצובר יש מספר השונה מאפס (דגל האפס נמצא ב-"0") אז D1 היה "1". בדיקת מצב הסיבית תהיה בהתאם לדגל האפס.

נכתוב תוכנית המוציאה לנוריות הפלט את מספר המפסק הנלחץ.
לדוגמא, לחיצה על המפסק הרביעי תגרום שיופיע בנוריות הפלט המספר 04H (ולא המספר 08H שהיה מופיע שם, אילו העברנו את מצב המפסקים ישירות אל הפלט).

ב-6800 קימת פקודה הנקראת BIT (BIT TEST), המבצעת את הפקודה. הלוגית AND על הצובר, אבל מבלי לשנות את הצובר. ב-8085 לא קיימת פקודה כזו, לכן יש לשמור את תוכן הצובר באוגר ולאחר כל בדיקה לטעון את הצובר מחדש.

תרשים זרימה א'





שים לב שהתוכנית בודקת שמונה מפסקים, בזה אחר זה בלולאה אינסופית.
חשוב מה קורה כאשר שני מפסקים מופעלים.

5.3 סריקת מפסקים רגעיים

מפסק רגעי (MOMENTARY SWITCH) הוא מפסק שיש לו מצב נורמלי מסוים (סגור או פתוח) ומצב רגעי - המשלים של המצב הנורמלי. בלחיצה על המפסק הוא עובר למצבו הרגעי ונשאר במצב זה עד לשחרורו. דוגמא למפסקים רגעיים הם מקשי מחשב. לצורך נוחיות, נניח שהמצב הנורמלי הוא מצב LOW והמצב הרגעי הוא מצב HIGH.

כדי להגיב לפעולת מקש, יש לבדוק עלית מקש ל-HIGH ולפעול בהתאם למקש הנלחץ. לחיצה על מקש במחשב, גורמת להדפסת אות מתאימה במסך. לחיצה כפולה על אותו מקש, גורמת להדפסת כפולה של אותה האות, בהמשך אותה שורה. פעולת המחשב, בתגובה לעליית המקש ל-HIGH, מסתיימת לפני ירידת המקש ל-LOW (המחשב מהיר מאצבע אדם והמקש המיכני). במקרה כזה, חוזר המחשב לסרוק את המקשים ומוצא את המקש הלחוץ עדיין ב-HIGH ושוב מגיב. כתוצאה מכך, נקבל שורה של תוים עקב לחיצה אחת.

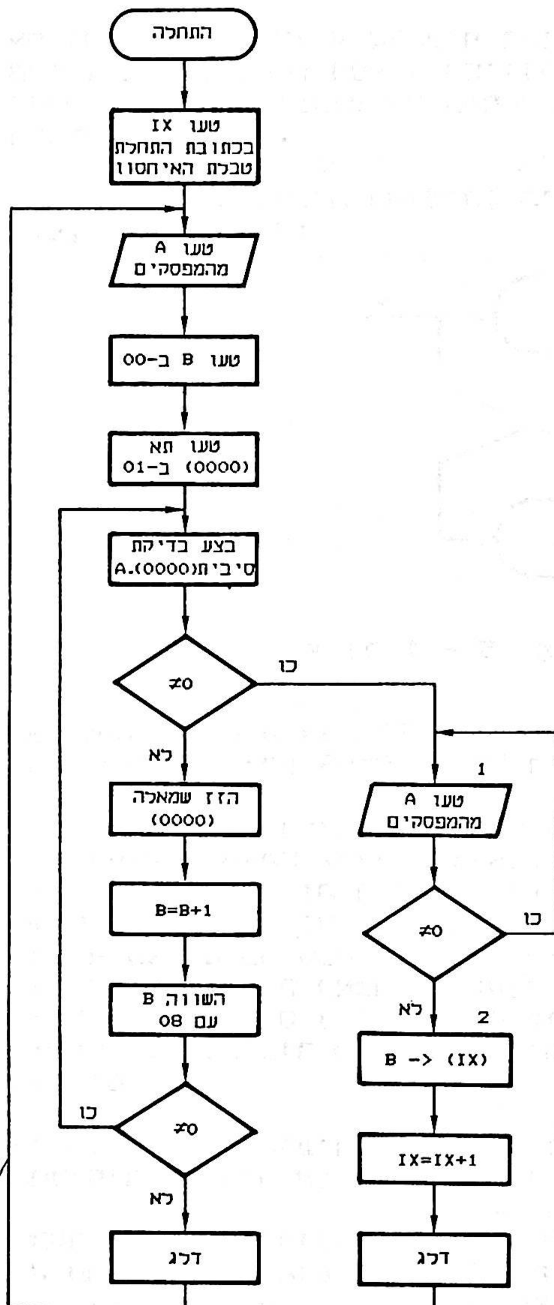
פתרון פשוט לבעיה (קיימים פתרונות נוספים) הוא לרשום את עלית המקש המסוים ל-HIGH ולהגיב רק לאחר ירידתו ל-LOW.

נשכלל את התוכנית מהסעיף הקודם. במקום להוציא את המספרים הסידוריים של המפסקים הנלחצים אל הנוריות, תאחסן אותם בשורת תאים בזכרון. התוכנית רצה בלולאה אינסופית.

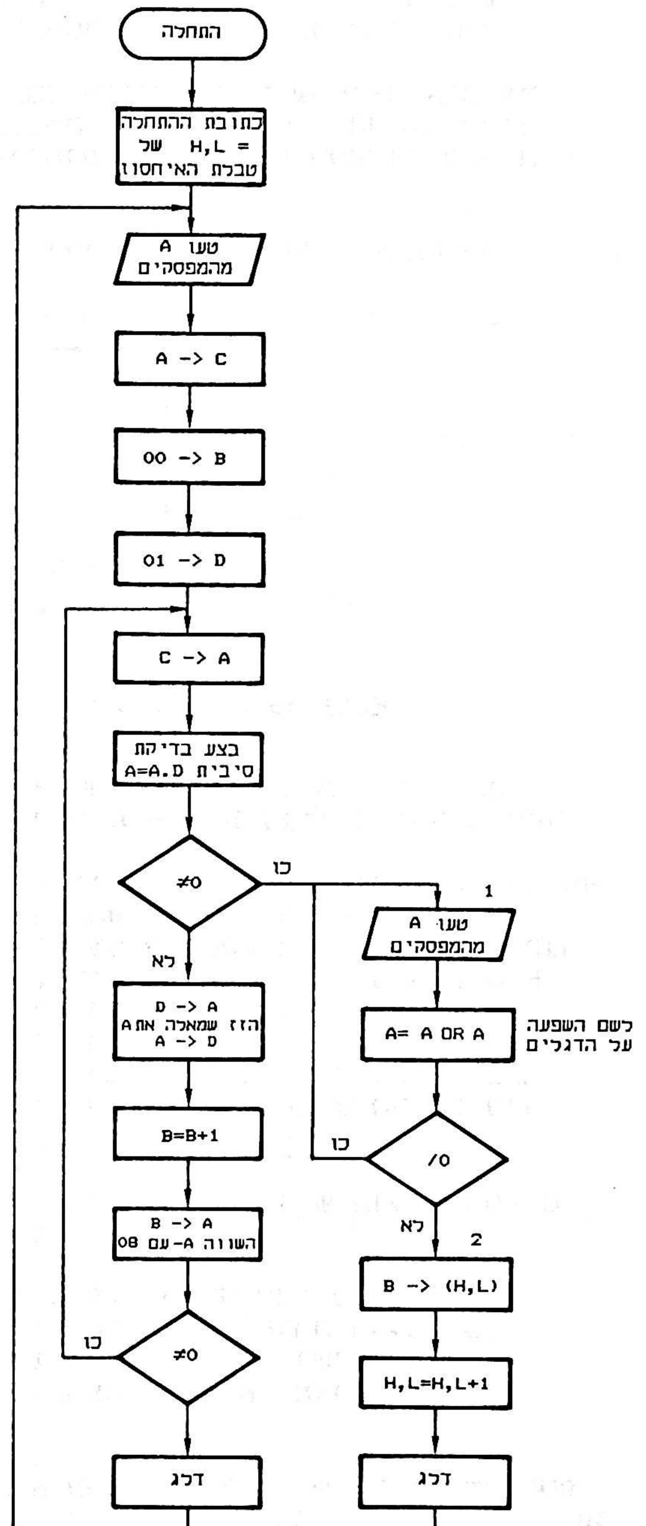
ניתן להוסיף תנאי כזה, שלחיצה על מפסק מסוים תגרום להפסקת התוכנית. במחשב, לחיצה על מקש C.R (CARRIAGE RETURN) גורמת למעבר לשורה חדשה ולפענוח השורה הרשומה.

לסימנים 1 ו-2 שבתרשים הזרימה נתייחס בסעיף הבא.

6800


התחלה
טען IX
בכתובת התחלת
טבלת האיחסון
טען A
מהמפסקים
טען B ב-00
טען תא
(0000) ב-01
בצע בדיקת
סיבית(0000).A
≠0
כן
לא
הזז שמאלה
(0000)
B=B+1
השווה B
עם 08
≠0
כן
לא
דלג
1
טען A
מהמפסקים
≠0
כן
לא
2
B -> (IX)
IX=IX+1
דלג

8085

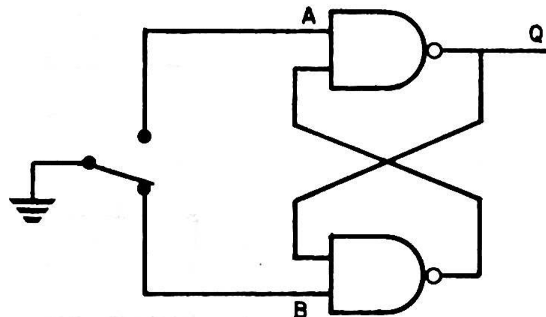

התחלה
כתובת ההתחלה
של H,L =
טבלת האיחסון
טען A
מהמפסקים
A -> C
00 -> B
01 -> D
C -> A
בצע בדיקת
סיבית A=A.D
≠0
כן
לא
D -> A
הזז שמאלה את A
A -> D
B=B+1
B -> A
השווה A-עם 08
≠0
כן
דלג
1
טען A
מהמפסקים
A= A OR A
לשם השפעה
על הדגלים
/0
כן
לא
2
B -> (H,L)
H,L=H,L+1
דלג

5.4 ביטול ריטוט המקשים (BOUNCING)

בכל מפסק קיימת בעית ריטוט, המופיעה בזמן מעבר של המפסק. לחיצה על המפסק, גורמת למגעיו לרטוט (להתחבר ולהתנתק) מספר פעמים עד להתייצבותם במצב החדש. גם שחרור המפסק גורם לתופעת הריטוטים.

אם נריץ את התוכנית שבסעיף הקודם כמו שהיא, נקבל שלחיצה אחת על מקש גורמת להדפסת מספרו הסידורי בטבלה מספר פעמים. המחשב מהיר יותר מזמן ההתייצבות של המפסק. קיימות שתי שיטות לפתרון הבעיה - בחומרה ובתוכנה.

השיטה לפתרון הבעיה בחומרה, היא על ידי חיבור FLIP - FLOP לכל מקש, בצורה הבאה:



ציור 1 - 5 מעגל לביטול ריטוט של מפסק

אם השערים הם מסוג TTL, הרי שהדק כניסה באויר מהווה "1". אם משתמשים בלוגיקת CMOS יש לחבר לכניסות A - ו-B נגדי משיכה ל-VCC.

בזמן מעבר, מתקבלים הריטוטים בין המגעים (A או B) למירווח, אבל לא בין המגעים עצמם (בין A ל-B). במצב המתואר בציור B ב-"0" ו-A ב-"1". Q במצב זה ב-"0". בזמן מעבר של המפסק, הדק B נפתח ונסגר חליפות. הדק A נשאר כל הזמן ב-"1", בזמן ש-B רוטט. למרות ש-B מחליף מצב מספר פעמים מ-"0" ל-"1" וחזרה ל-"0", Q נשאר ללא שינוי (בדוק זאת). רק כאשר A מתקצר לאדמה בפעם הראשונה, Q מחליף מצב ל-"1". ריטוטים נוספים ב-A שוב לא יגרמו לשינוי ב-Q והוא ישאר ב-"1" עד לקיצור B לאדמה שנית. כל זאת בזכות המשוב הקיים בין השערים.

לפתרון זה של הבעיה, יש מחיר גבוה. הפתרון מחייב שימוש במפסקים מחליפים וחיבור מעגל אלקטרוני לכל מקש.

לפתרון הבעיה בתוכנה, קיימות שתי שיטות. שיטה אחת מבוססת על לקיחת שורת דגימות מהמקשים. רק כאשר מבחינים שסדרה רצופה של דגימות נותנת אותו המצב מתייחסים לשנוי מצב זה. בשיטה זו, התייחסות היא רק למצבים היציבים של המפסקים ולא למצבי הריטוט שלהם.

השיטה השניה פשוטה יותר, אבל גם היא אמינה למדי. שיטה זו מבוססת על הוספת לולאת השהיה לאחר כל הבחנה בשינוי מצב למשך זמן ארוך יותר

מזמן הריטוטים.שיטה זו מחייבת הכרת משך זמן הריטוטים של המפסק. אין טעם לתת זמן ארוך מדי, כי אז לא יובחנו לחיצות מהירות על המקשים.

את לולאות ההשהיה יש להוסיף לתרשים זרימה ב' במקום המסומנים ב- 1 וב- 2.

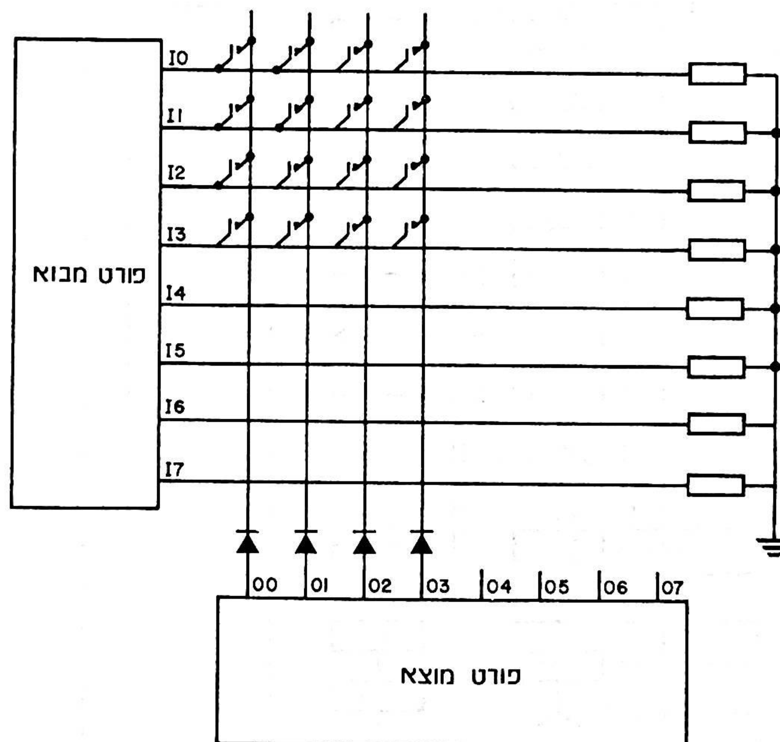
5.5 לוח מקשים - KEYBOARD

בשיטות שתוארו לסריקת המקשים, חיברנו כל מפסק אל סיבית בפורט. הדבר מחייב שימוש בפורטי מבוא כמספר המקשים חלקי שמונה.

לוח מקשים מורכב ממספר רב של מפסקים ושיטה זו של סריקה בזבזנית למדי בחומרה. למשל, 64 מקשים (לוח מקשים מלא) מחייב שימוש בשמונה פורטים.

כדי להתגבר על בעיה זו, עורכים את לוח המקשים בצורה של מטריצה. משתמשים בפורט מוצא ובפורט מבוא. לדוגמא נתאר מטריצה של 16 מקשים.

להפעלת מטריצה זו, זקוקים ל-4 סיביות של פורט מוצא ו-4 סיביות של פורט מבוא. ניתן גם להשתמש ב-2 סיביות פורט מוצא ו-8 סיביות פורט מבוא ולהיפך. נשתמש בצורת הארגון הראשונה.



ציור 2 - 5 ארגון 16 מקשים במטריצה

עקרון סריקת המקשים מבוסס על האלגוריתם הבא:

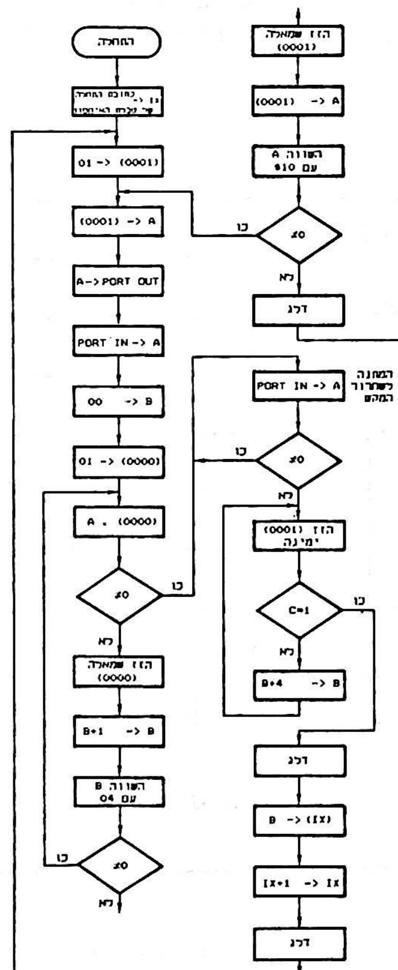
א. הוצא "1" לקו O0 של פורט המוצא.
ב. סרוק את קוי הכניסה של פורט המבוא לפי התוכנית של תרשים זרימה ב'. אם כל המקשים בשורה זו אינם לחוצים, כל הכניסות תהיינה ב"0".
ג. אם אחד המקשים בשורה זו לחוץ יתקבל "1" בקו שלו, עבור לסעיף ו'.
ד. הוצא "1" לקו Q1 של פורט המוצא.
ה. חזור על סעיפים ב' ו-ג'.
ו. מספר המקש הנלחץ הוא מספר הקו בפורט המוצא המחובר למקש הנלחץ כפול 8 פלוס מספר הקו בפורט המבוא שם נמצא "1".

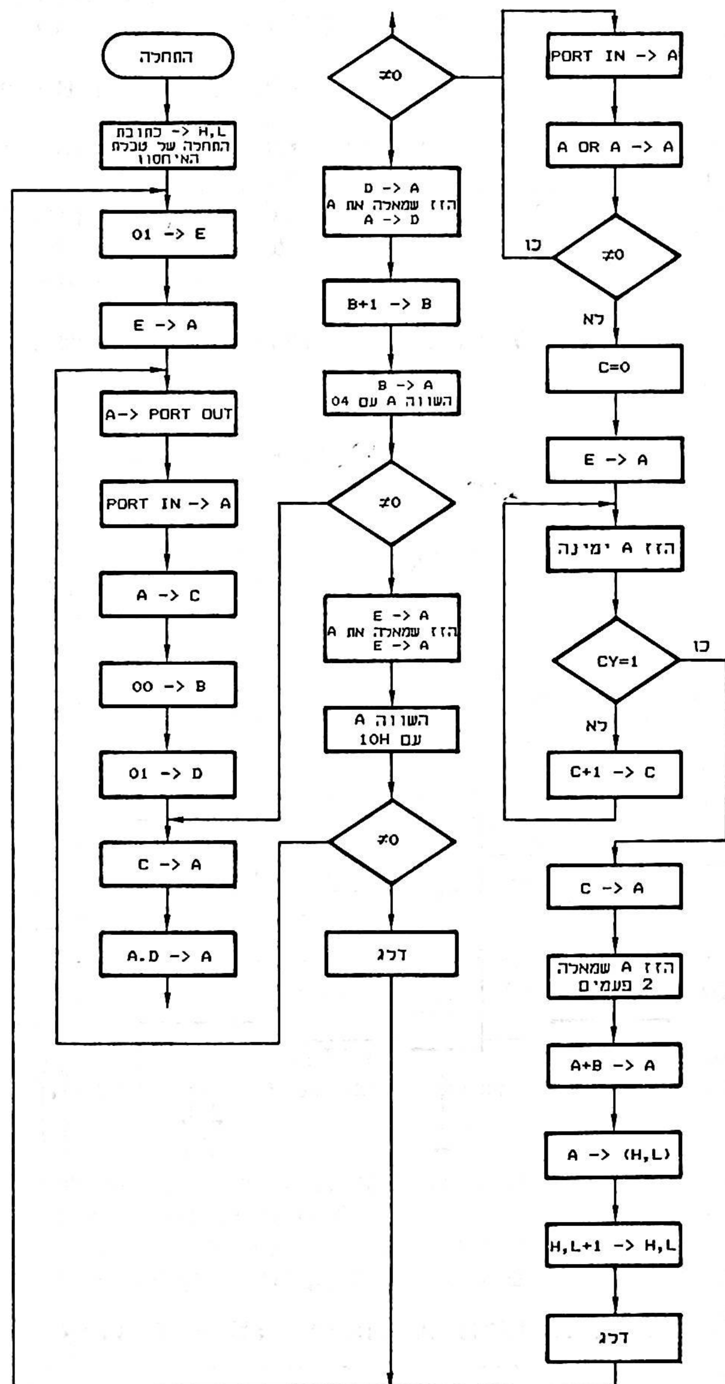
בשרטוט הנראה בציור 2-5 קל לראות את אופן הרחבת לוח המקשים לעד 64 מקשים, על ידי הוספת שורות מקשים נוספות.

ניתן לחסוך בסיביות פורט מוצא על ידי שימוש במפענח (DECODER). לדוגמא, כדי לקבל 8 שורות מפסקים, מחברים 3 סיביות מפורט המוצא למפענח "1 מ-8", ואת יציאות המפענח נחבר לשורות.

לפניך תוכניות המבוססות על תרשים זרימה ב' ומבצעות את האלגוריתם הזה.

תרשים זרימה ג' ל - 6800





התחלה
כתובת -> H,L
התחלה של טבלת
האיחסון
01 -> E
E -> A
A-> PORT OUT
PORT IN -> A
A -> C
00 -> B
01 -> D
C -> A
A.D -> A
≠0
D -> A
הזז שמאלה את A
A -> D
B+1 -> B
B -> A
השווה A עם 04
≠0
E -> A
הזז שמאלה את A
E -> A
השווה A
עם 10H
≠0
דלג
PORT IN -> A
A OR A -> A
≠0
כן
לא
C=0
E -> A
הזז A ימינה
CY=1
כן
לא
C+1 -> C
C -> A
הזז A שמאלה
2 פעמים
A+B -> A
A -> (H,L)
H,L+1 -> H,L
דלג

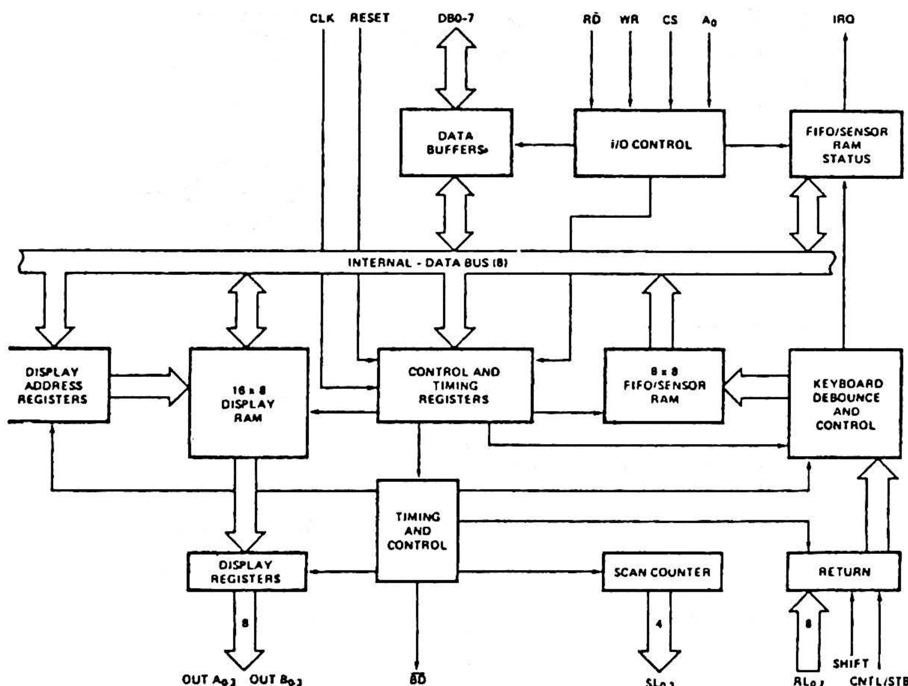
PROGRAMMABLE KEYBOARD/DISPLAY INTERFACE - 8279 5.6

ה-8279 הוא רכיב המתחבר אל BUS ה-CPU ומאפשר הצגת תוים בתצוגת ה-7-SEG. וסריקת המקשים בו זמנית. הוא מבצע את הצגת התווים וסריקת המקשים בחומרה מבלי להיעזר ב-CPU לשם כך. בצורה זו הוא משחרר את ה-CPU לריצה על תוכניות אחרות.

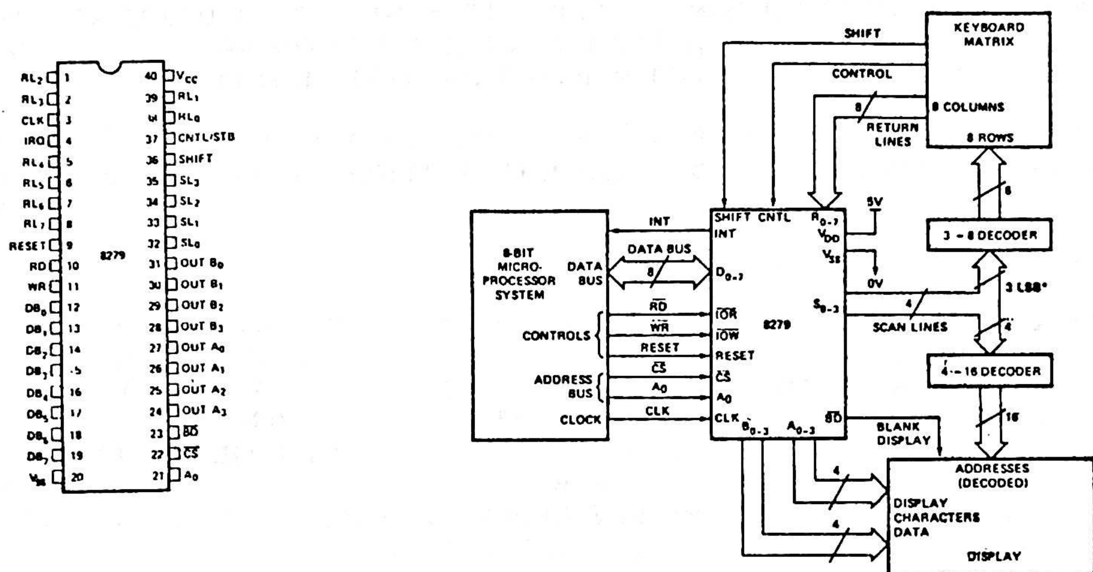
השימוש ברכיב מפשט במידה ניכרת את תוכנת המערכת.

הרכיב תופס שתי כתובות במפת הזכרון או במפת I/O בהתאם למעגל הפענוח. מעגל הפענוח מפענח את קוי הכתובת מלבד את קו A0. קו A0 מתחבר ישירות לרכיב. בצורה זו מתקבל CS בפניה לשתי כתובות. הפורט הגבוה (כאשר A0=1) משמש להעברת COMMAND או STATUS והפורט הנמוך להעברת DATA בהתאם.

דיאגרמת המלבנים של הרכיב וצורת השמוש בו מתוארים בציורים הבאים:



ציור 3 - 5: דיאגרמת מלבנים של ה-8279.



*Do not drive the keyboard decoder with the MSB of the scan lines.

ציור 4 - 5 אופן חיבור ה-8279 ללוח מקשים ולתצוגה.

ל- 8279 מצבי הפעלה שונים ומגוונים. נתאר רק שניים מהם. השאר ניתן למצא בדפי הנתונים של הרכיב.

א. פניה לתצוגה

ל-8279 RAM פנימי בן 16X8 BIT המשמש את התצוגה. הוא מציג את תוכן תאי ה-RAM בצורה דינמית, בזה אחר זה בתצוגה ה-.7-SEG יחידות ה-.7-SEG מחוברות כמתואר בפרק הקודם. בכל יחידת תצוגה יש 8 לדים (7-SEGMENTS ונקודה עשרונית). 8 הלדים מחוברים במקביל. כניסות ה-COMMON של יחידות התצוגה מחוברות ליציאות מפענח "1 מ-16" של קוי ה-.S.L (SCAN LINES). הרכיב, בעזרת קוי ה-.S.L, פונה כל פעם ליחידת תצוגה אחרת ומעביר אליה מה-RAM את תוכן התא המתאים.

כל סיבית בנתון המועבר, מדליקה או מכבה (1- מכבה, 0-מדליק) את הלד המתאים. לדוגמא, הנתון FB (11111011) מדליק את הקו האמצעי בלבד.

ניתן לחבר אל הרכיב תצוגה שמספר יחידותיה קטן מ-16, על ידי שימוש בחלק מיציאות המפענח. אי אפשר להשתמש ברכיב לתצוגה בעלת יותר מ-16 יחידות.

כדי להציג נתון מסוים באחת מיחידות ה-.7-SEG, יש לפנות לפורט הגבוה ולהעביר אליו את הנתון 1 0 0 A1 A A A A . AAAA מציין את כתובת התא אליו פונים. לאחר מכן פונים לפורט הנמוך ומוסרים את הנתון שרוצים שיכנס לתא הנ"ל. ה-8279 יציג כל הזמן את תוכן התאים בתצוגה.

כאשר מעבירים שורה של נתונים לתאי ה-RAM, בזה אחר זה אין צורך לפנות לפורט הגבוה בכל פעם (לשם העברת כתובת תא חדש' מספיק שבפעם

הראשונה, נכניס בסיבית A1 - "1" לוגי. לאחר כל פניה לפורט הנמוך, תגדל הכתובת AAAA אוטומטית ב-1. אם A1=0, אזי הכתובת אינה משתנה אלא לאחר פניה לפורט הגבוה עם כתובת אחרת.

לדוגמא, תוכנית המדליקה את הקו האמצעי ב-4 מיחידות התצוגה. נניח, שמעגל הפענוח קובע את כתובת הפורט הגבוה ל-05 וכתובת הפורט הנמוך ל-04.

6800	8085	
START: LDAA#$90	START: MVI A,90H	טען A ב-10010000
STAA $8005	OUT 05H	העבר A לאוגר הבקרה
LDAA#$FB	MVI A,FBH	טען A ב-11111011
LDAB#$04	MVI B,04H	טען מונה (B) ב-04
LOOP1: STAB $8004	LOOP1: OUT 04H	הוצא תוכן A לפורט נמוך
DCRB	DCR B	חסר 1 מהמונה
BNE LOOP1	JNZ LOOP1	אם שונה מאפס דלג ל-LOOP1.

התוכנית אינה מסתיימת כי היא יכולה לשמש כחלק מתוכנית כללית.

ב. פניה למקשים

כמתואר בציור 4-7, יש ל-8279 קו בקשת פסיקה. קו זה מחברים לאחד מקוי הפסיקה של ה-CPU בהתאם לעדיפות שרוצים להעניק לו. את המקשים מחברים במטריצה אל קוי ה-RL (RETURN LINES) מצד אחד ואל יציאות מפענח "1 מ-8" מצד שני. המפענח מפענח את שלשת קוי ה-LSB של קוי ה-SCAN LINES של ה-8279 (ראה ציור). סך הכל, ניתן לחבר מטריצה של 8X8 מקשים.

ה-8279 סורק את המקשים של הזמן, בלי תלות בעבודת ה-CPU. לחיצה על אחד המקשים גורמת ליצירת בקשת פסיקה על ידי הרכיב, והכנת מספר המקש הנלחץ בפורט הנמוך.

תוכנית הפסיקה של ה-CPU צריכה לפנות לפורט הנמוך ולקחת משם את מספר המקש.

5.7 שאלות הכנה

1) המר את תרשים זרימה א' לתוכנית, כאשר המפסקים מחוברים לפורט IN1-1 בכתובת IO-04.
כתוב את התוכנית בהתאם לסוג המיקרופרוססור שברשותך.

2) המר את תרשים זרימה ב' לתוכנית. המפסקים מחוברים כמו בסעיף 1).

3) הוסף לתוכנית שבסעיף 2) לולאות השהיה במקומות המסומנים ב-1 וב-2 בתרשים הזרימה.

4) המר את תרשים זרימה ג' לתוכנית, כאשר גם פורט IN וגם פורט OUT הם בכתובת IO-04.

5) שרטט מעגל חשמלי מפורט בהתאם למעגל המתואר בציור 4-4.

מהלך הניסוי

5.8 זיהוי מפסק נלחץ

מטרת הניסוי היא לקרא את מצב מפסקי הערכה ולזהות את המפסק הנלחץ, על ידי העברת מספרו הסדורי לנוריות הפלט. כל זאת בעזרת התוכנית לתרשים זרימה א'.

1) חבר את מפסקי הערכה ל- PORT IN-1.

2) חבר את יציאות PORT OUT-1 לכניסות ה"לדים" של הערכה.

3) קבע כתובות הפורטים בעזרת מעגל הפענוח.

4) כתוב תוכנית לתרשים זרימה א'.

5) הרץ את התוכנית בלולאה אינסופית. שנה את מצב המפסקים ובדוק אם מספר המפסק המורם אכן מופיע בנוריות הפלט.

6) בדוק ורשום מה קורה כאשר שני מפסקים מורמים.

5.9 סריקת מפסקים רגעיים

מטרת הניסוי היא לקרוא את מצב מפסקי הערכה ולזהות את המפסק הנלחץ כמו בניסוי הקודם, כאשר במקום להעביר את מספרי המפסקים הנלחצים לנוריות, להעבירם לטבלה בזה אחר זה, בהתאם לתרשים זרימה ב', ולזהות את בעית הריטוט (BOUNCING).

1) חבר ל- PORT IN-1 ול- PORT OUT-1 את המפסקים וה"לדים" בהתאמה.

2) קבע את כתובות הפורטים בעזרת מעגל הפענוח.

3) כתוב תוכנית לתרשים זרימה ב', כאשר הנך קובע כתובת התחלתית לטבלה שלשם תעביר התוכנית את מספרי המקשים הנלחצים.

4) הרץ את התוכנית בלולאה אינסופית.

5) הרם בכל פעם מפסק אחד והורד אותו חזרה. רשום לעצמך את סדר הרמת המפסקים.

6) הפסק את התוכנית. פנה לטבלה ובדוק אם המספרים הרשומים שם מתאימים לסדר הלחיצה על המפסקים.

7) יש סכוי סביר, שתקבל על כל לחיצת מפסק, את מספרו הסדורי מספר פעמים. הסבר מדוע.

5.10 ביטול ריטוט מפסקים בתוכנה

מטרת התוכנית היא למצוא את אורך לולאות ההשהייה הדרושות כדי לבטל את תגובת המחשב לריטוט המפסקים, והוספת לולאות השהייה בהתאם לתוכנית מהניסוי 5.9.

1) שפר את התוכנית מניסוי 5.9 על מנת לבטל את התופעה שתוארה בסעיף 7). הוסף לולאת השהייה. התחל בלולאות השהייה קצרות מאד.

2) הרץ את התוכנית ובדוק אם המחשב מגיב עדיין לריטוט המפסקים.

3) אם כן, הגדל את ההשהייה וחזור על סעיף 2), עד לביטול תגובת המחשב לריטוט המפסקים.

4) חשב את גודל ההשהייה הנדרשת ב-US ורשום אותה.

5.11 חיבור לוח מקשים במטריצה

מטרת הניסוי היא לחבר מטריצה של 16 מקשים אל פורטי המערכת וזיהוי המקשים הנלחצים בהתאם לתוכנית של תרשים זרימה ג'.

1) חבר את PORT OUT-1 אל ה"לדים" בערכה.

2) חבר מטריצה של 16 מקשים המאורגנים בצורה של 4X4 אל PORT OUT-2 ואל PORT IN-1 בהתאם לציור 5-2.

3) המר תרשים זרימה ג' לתוכנית. שנה אותו כך, שיחד עם העברת מספר המקש הנלחץ לטבלה, יוציא אותו אל נוריות הפלט. התייחס לכתובות הפורטים בהתאם למעגל הפענוח.

4) הרץ את התוכנית.

5) לחץ על אחד המקשים ובדוק אם מספרו הסדורי מופיע בנוריות הפלט.

6) לחץ על מספר מקשים ורשם לך את סדר לחיצתם.

7) הפסק את התוכנית. פנה לטבלת האיחסון ובדוק אם מספרי המקשים מופיעים שם.

8) בדוק אם תופעת התגובה לריטוט חזרה על עצמה.

9) שפר את תרשים זרימה ג' והתוכנית כדי לבטל את תגובת המחשב לריטוט.

10) חזור על סעיפים 5) עד 8).

5.12 חיבור תצוגה ולוח מקשים באמצעות 8279

מטרת הניסוי היא לחבר את תצוגת ה- 7-SEG. של הערכה ולוח מקשים אל BUS המיקרו-מחשב בעזרת הרכיב 8279 והפעלתם.

1) הפוך את המעגל המתואר בציור 5-4 למעגל חשמלי מפורט.

2) בנה את המעגל החשמלי על לוח הניסויים בעזרת ה-8279 ושני מפענחים. את היציאות לתצוגה חבר לכניסות ה- 7-SEG. בהתאם. את היציאות ללוח מקשים חבר ללוח המקשים המאורגן במערך של 4X4.

3) חבר את קו A0 אל ה-8279.

4) חבר את קו CS לכניסת ה-CS של הרכיב. קו CS מפענח את הקוים A7-A2, כלומר קו A1 לא מחובר כלל. בצורת חיבור זו, יתקבל CS בפניה לכתובת מתאימה לפי A2-A7, גם כאשר A1 ב- "1" וגם כאשר A1 ב- "0". אין זה מפריע למהלך הניסוי. כתובת הפורט הגבוה וכתובת הפורט הנמוך ב-8279 יקבעו בהתאם ל-A0.

5) קבע את מעגל הפענוח לתחום הכתובות 04 ו-05.

6) את קו הפסיקה חבר לקו פסיקה המיועד למשתמש במערכת המיקרו-מחשב שברשותך.

7) הרץ את התוכנית שבסעיף 4.6 - א). סיים את התוכנית בעצור. בדוק אם נדלק הקו האמצעי בכל ארבעת יחידות התצוגה.

8) כתוב תוכנית פסיקה האוספת את הנתון מפורט 04 (הפורט הנמוך) ומוציאה אותו אל התצוגה.

9) הרץ תוכנית ראשית כלשהי. לחץ על מקש ובדוק את המתקבל בתצוגה.

10) שנה את תוכנית הפסיקה כך שתאסוף את הנתון, תמיר אותו בעזרת טבלת חיפוש מבסיס 16 ל- 7-SEG. ותוציא אותו אל התצוגה. היעזר בתוכניות הרשומות בפרק הקודם. חזור על סעיף 9).

פרק 6 - D TO A ו- A TO D

6.1 נושאי הפרק

א. הכרת DIGITAL TO ANALOG (D/A) בעזרת מגבר שרת + רשת נגדים.
ב. בנית ANALOG TO DIGITAL (A/D) בעזרת D/A.
ג. הכרת השימוש ב-A/D מרובב עם מספר כניסות אנלוגיות.

רקע עיוני

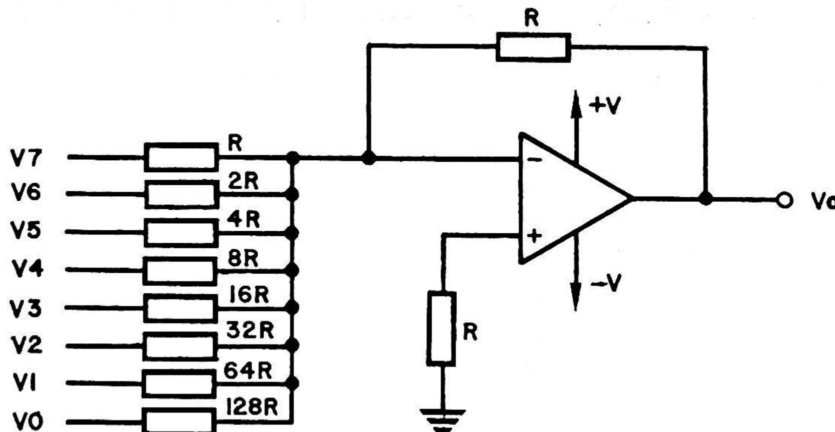
6.2 D TO A בעזרת מגבר שרת ורשת נגדים

מעגל DIGITAL TO ANALOG הוא מעגל המקבל בכניסותיו מספר בינארי כלשהו ומוציא ביציאותיו מתח חשמלי. המספר הבינארי מורכב מספרות בינאריות (סיביות) שיכולות לקבל ערך לוגי "0" או "1", כאשר "0" שווה ל-0V ו-"1" שווה למתח ייחוס מסוים - VR.

המתח במוצא המעגל שווה לערך העשרוני של המספר הבינארי מוכפל ב-VR ומוכפל במקדם שהוא פונקציה של המעגל. לדוגמא, נניח שהמספר הבינארי הוא 01001100 השווה ל-78, מתח הייחוס הוא 5V ומקדם המעגל הוא 0.01. במוצא המעגל יתקבל המתח

$$VO = 0.01 \times 5 \times 78 = 3.9V$$

בנית מעגל D/A נעשית בדרך כלל בעזרת מגבר שרת + רשת נגדים. בשיטה זו בונים למעשה מגבר מסכם כמתואר בציור 6-1.



ציור 6-1: מעגל D/A בעזרת מגבר שרת ורשת נגדים בעלי שקלול בינארי

נחשב את מתח המוצא VO כפונקציה של V7 - VO.

$$VO=-\left(\frac{R}{R}V7+\frac{R}{2R}V6+\frac{R}{4R}V5+\frac{R}{8R}V4+\frac{R}{16R}V3+\frac{R}{32R}V2+\frac{R}{64R}V1+\frac{R}{128R}VO\right)$$

נצמצם ב- R

$$VO=-\left(V7+\frac{1}{2}V6+\frac{1}{4}V5+\frac{1}{8}V4+\frac{1}{16}V3+\frac{1}{32}V2+\frac{1}{64}V1+\frac{1}{128}VO\right)$$

נכפיל ונחלק ב - 128

$$VO=\frac{1}{128}(128V7+64V6+32V5+16V4+8V3+4V2+2V1+VO)$$

אם כל כניסת מתח שווה למתח היחוס VR כפול ערך הסיבית המחוברת אליה ("0"או"1")

$$VI = VR.DI$$

אזי נקבל

$$VO=\frac{1}{128}(128VR.D7+64VR.D6+32VR.D5+16VR.D4+8VR.D3+4VR.D2+2VR.D1+VR.DO)$$

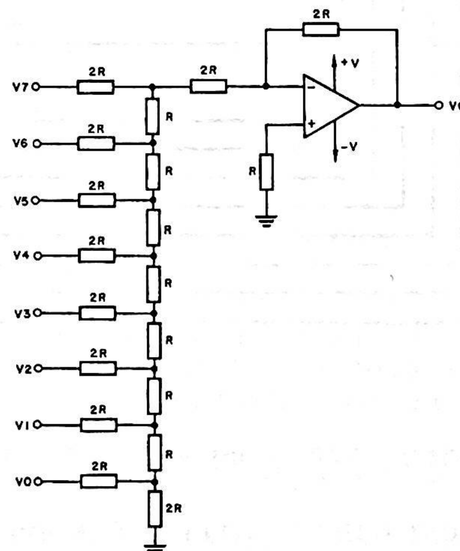
$$VO=\frac{1}{128}VR(128D7+64D6+32D5+16D4+8D3+4D2+2D1+1DO)$$

בצורה כזו מתקבל מתח המוצא שערכו המוחלט הוא פונקציה של המספר הבינארי מחובר לכניסות המעגל. המקדם במעגל זה שווה ל - $\frac{1}{128}$.

המתח במוצא הוא מתח שלילי. כדי לקבל מתח חיובי ניתן לחבר למוצא המעגל מגבר מהפך. ניתן לתת לו גם הגבר מסוים כדי לשנות את המקדם הכללי של המעגל.

במעגל D/A השיבות מרובה לדיוק הנגדים. הבעיה במעגל זה היא השמוש ברשת נגדים שונים החייבים לשמור על ערך יחסי מדויק זה מזה, כמתואר בציור. בנית מעגל מדויק כזה היא קשה ויקרה יחסית.

פתרון לבעיה זו ניתן למצא במעגל המתואר בציור 2-6. מעגל זה משתמש ברשת של נגדים זהים (2 סוגים בלבד). רשת זו מכונה רשת סולם.



ציור 2-6: מעגל D/A עם מגבר שרת ורשת נגדים מסוג סולם של R ו-2R.

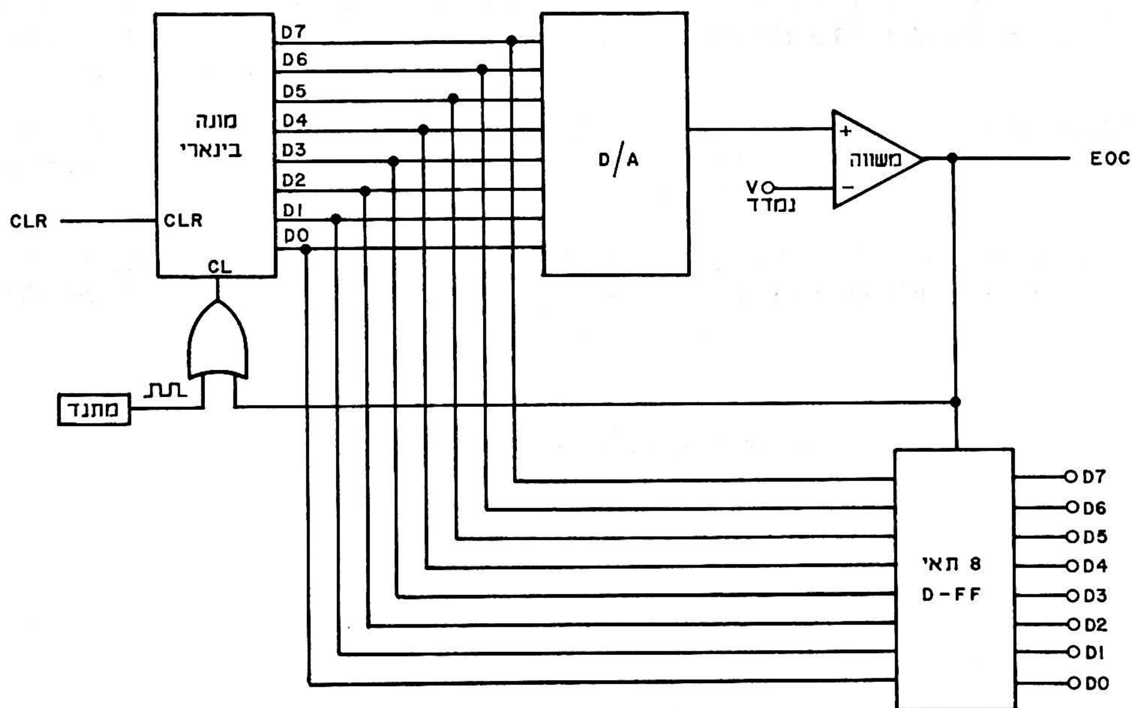
מצא את הקשר בין מתח המוצא למתחי המבוא. לנוחיותך השתמש בשיטת הסופרפוזיציה.

6.3 A TO D בעזרת D TO A

מעגל ANALOG TO DIGITAL הוא מעגל בעל כניסת מתח אנלוגי וקוי מוצא בינאריים. מעגל זה מוציא בקוי המוצא שלו את הערך הבינארי של מתח הנמדד שבכניסתו כפונקציה של מתח ייחוס מסוים.

מעגל A/D כולל בדרך כלל, מגבר משווה, שלכניסה אחת שלו מחובר המתח הנמדד. לכניסה השניה מספקים מתח הגדל בצורה הדרגתית כפונקציה של מספר בינארי גדל. ברגע שאות זה גדול מהמתח הנמדד, מחליף המגבר המשווה מצב ועוצר את העלאת המתח. שנוי זה גם גורם להוצאת המספר הבינארי שהתקבל ליציאות ה- A/D.

אחת השיטות לקבלת אות הגדל בצורה הדרגתית היא שיטה הנעזרת ב-D/A שתואר בסעיף הקודם. ראה ציור 6-3.



ציור 3 - 6 : מעגל A/D בעזרת D/A.

ההמרה מתחילה במתן פולס CLR למונה. איפוס המונה גורם להוצאת 0V ביציאת ה-D/A שגורם לירידת קו EOC ל-LOW. ירידה זו מאפשרת העברת פולסים למונה שמתחיל למנות.

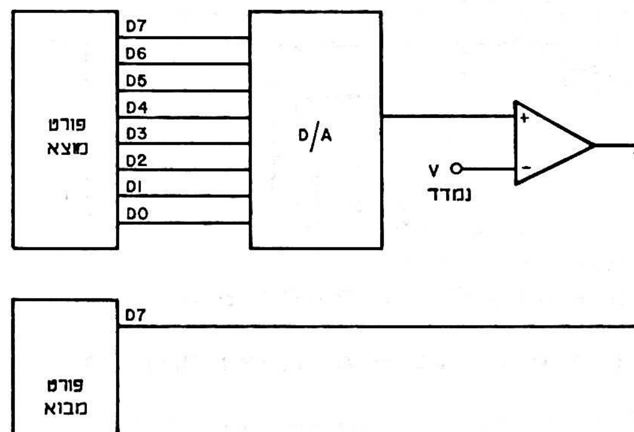
המספרים הבינאריים הגדלים שביציאות המונה, גורמים להופעת מתח גדל בצורה הדרגתית כניסה הלא מהפכת של המגבר המשווה. ברגע שמתח זה עובר את המתח הנמדד, עולה קו EOC ל-HIGH.

עליה זו עוצרת את העברת הפולסים למונה ומסיימת את הספירה, וכן גורמת להכנסת המספר הבינארי שביציאות המונה לתאי ה-D-F.F. קו EOC משמש גם כקו המסמן למערכת המתחברת ל-A/D שההמרה הסתיימה (END OF CONVERSIOR).

אם רוצים לחבר A/D כזה למיקרו-מחשב יש לחבר את יציאותיו לפורט מבוא או לחוצץ שיצא מ-3-STATE רק עם פנית ה-CPU אליו.

אנו רואים שהמרה מתקבילי לספרתי מתרחשת במשך זמן מסוים. זו הסבה שלכל מערכת A/D יש מלבד כניסת מתח נמדד ויציאות בינאריות, קו כניסה המתחיל את ההמרה וקו יציאה המודיע על סיום תהליך ההמרה. אין לדגום את יציאות ה-A/D לפני תום תתהליך המרה.

ניתן לוותר על חלק ממרכיבי ה-A/D המתואר, ע"י המרת חומרה בתוכנה. במקום לחבר מונה, מתנד ותאי F.F. מחברים את כניסות ה-D/A ליציאותיו של פורט OUT ויוצרים את המספר הבינארי הגדל בתוכנה. מעגל זה יראה בצורה הבאה.



ציור 4 - 6 : A/D באמצעות D/A ותוכנה.

מוציאים לפורט המוצא מספרים בינארים גדלים בזה אחר זה, המתחילים מ-0. לאחר הוצאת כל מספר לפורט בודקים את סיבית D7 בפורט המבוא. ברגע שמוצאים שסיבית זו עלתה ל-"1" מפסיקים את הספירה. המספר האחרון שנשלח הוא הערך הבינארי המבוקש.

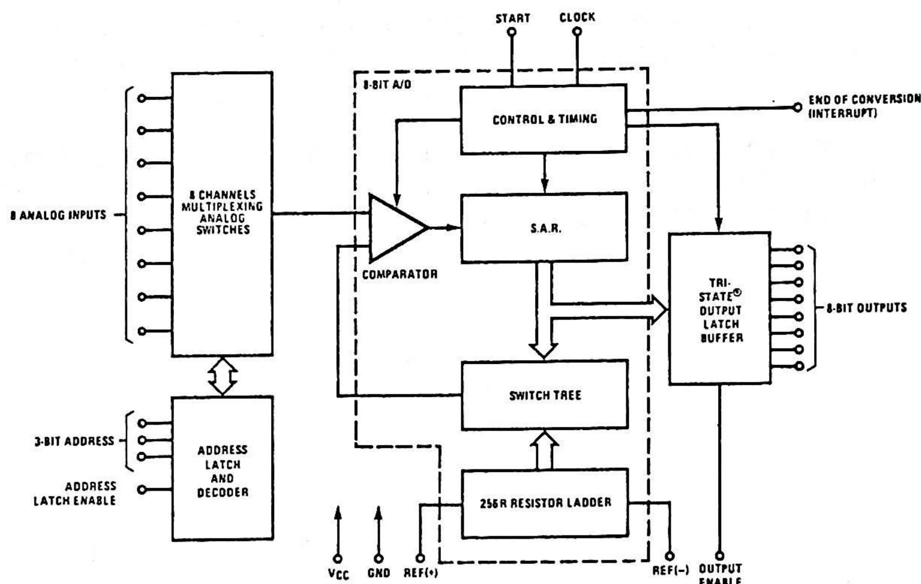
6.4 A/D מונוליתי עם 8 ערוצי כניסה

אין לסיים דיון ב-A/D מבלי לתאר את רכיבי ה-A/D המונוליתיים המותאמים למיקרו-מחשב. רכיבים אלה כוללים בתוכם את כל מעגל ההמרה, המתואר בציור 6-3, כולל חוצץ ביציאה. כל שהם זקוקים לשם

הפעלתם. הוא פולסי שעון בתדר מתאים ופולס START להתחלת ההמרה. בסיום ההמרה הם מוציאים את EOC. פניה לחוצץ היציאה גורמת להוצאת הנתון לקוי המוצא.

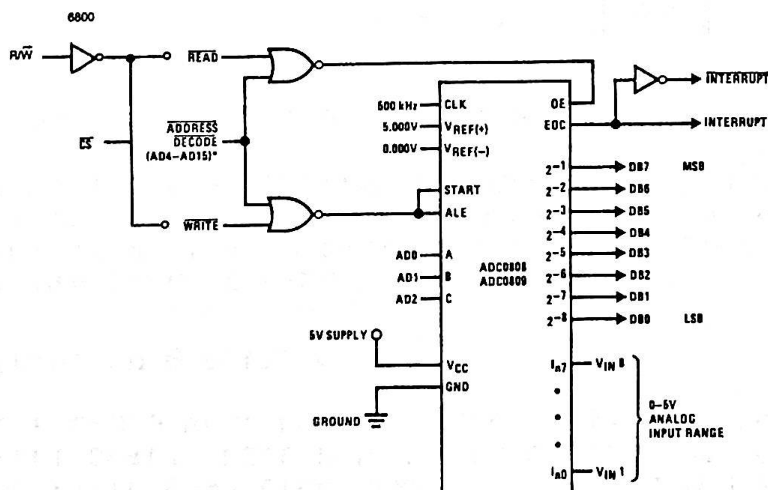
היום מקובל להשתמש ברכיבים מתוחכמים יותר מ-A/D למתח כניסה בודד. משתמשים ברכיב המרבב (MULTIPLEX) מספר כניסות. נתאר רכיב המרבב 8 כניסות אנלוגיות. לרכיב זה יש, מלבד הקוים הרגילים של A/D, 3 כניסות כתובת. ניתן לחבר אליו 8 כניסות אנלוגיות שונות. פניה לרכיב עם כתובת מתאימה ב-3 קוי כתובת אלה, ממירה מתקבילי (ANALOG) לספרתי (DIGITAL) את הערוץ שזוהי כתובתו.

רכיב כזה הוא ה-ADC0808 של NATIONAL והוא מתואר בציור 5-6.



ציור 5 - 6 : תיאור ה - ADC0808

חיבור רכיב כזה נעשה כמתואר בציור 6 - 6.



ציור 6 - 6 : אופן חיבור ADC0808 ל-BUS המיקרו-מחשב.

קו CS מתקבל מפענוח הקוים A15 - A3 כאשר עובדים במיפוי זכרון, או מפענוח הקוים A7 - A3 כאשר עובדים במיפוי IO. את הקוים A0,A1,A2 מחברים ישירות לרכיב. צורה זו של חיבור תיצור CS ב-8 כתובות רצופות שונות. כתיבה באחת הכתובות האלה (לא משנה עם איזה נתון) תבצע עליה ל-HIHG בכניסת ה-START של הרכיב וההמרה תתחיל. ההמרה תתבצע רק על הכניסה האנלוגית שמציינים הקוים A0,A1,A2.

כאשר הרכיב מסיים את ההמרה, עולה קו EOC ל-HIGH, ניתן לחבר אותו, כמו בציור, לקו פסיקה (בלוגיקה חיובית או שלילית), וכן ניתן לחבר אותו לסיבית בפורט IN. כאשר ה-CPU מבחין בעליה זו של קו EOC (בתגובה לבקשת הפסיקה או בדגימה בלולאה אינסופית של קו זה), הוא קורא מהכתובת שכתב בה את הנתון. הקריאה מוציאה מ-3-STATE את חוצץ המוצא.

רכיב זה הוא מספיק משוכלל כך שניתן לחבר אותו ישירות לקוי מסלול הנתונים ללא צורך בפורט כלשהו. כניסת ה - ALE מאפשרת לחבר אותו ישירות גם ל-8085 ללא צורך בהפרדת קוי הכתובת מקוי הנתונים.

6.5 שאלות הכנה

1) שרטט מעגל חשמלי מפורט של D/A בהתאם לציור 5-1. למעגל המוצא חבר מגבר מהפך בעל הגבר של 1/3- והתנגדות כניסה של 2KOHM. השתמש במגבר שרת 741 כאשר R הוא בעל ערך של 2KOHM.

2) שרטט מעגל חשמלי מפורט של D/A בהתאם לציור 5-2. למעגל המוצא חבר מגבר בעל הגבר של 1/4- והתנגדות כניסה של 5KOHM. השתמש במגבר שרת 741 כאשר R הוא בעל ערך של 5KOHM.

3) שרטט מעגל חשמלי מפורט של A/D בהתאם לציור 5-3.
כ-D/A השתמש במעגל של סעיף 2).
כמונה 8 סיביות השתמש ברכיב 4040.
כרכיב בעל 8 תאי .D-F.F השתמש ברכיב 74LS374.
בנה את המתנד בעזרת 555.

4) שרטט מעגל חשמלי מפורט של A/D בהתאם לציור 5-4.
כ-D/A השתמש במעגל של סעיף 2).

5) שרטט מעגל חשמלי מפורט כולל מעגל פענוח לרכיב ADC0808. חבר את יציאותיו ישירות למסלול הנתונים. את יציאת EOC חבר לסיבית D0 של פורט IN-1.

6) כתוב תוכנית הפונה לכתובת ערוץ 3 ב-A/D שחיברת בסעיף הקודם לשם התחלת ההמרה, לאחר מכן בודקת בלולאה אינסופית את סיבית D0 בפורט IN-1. ברגע שהיא מבחינה בעלית קו זה ל-HIGH היא קוראת מכתובת ערוץ 3 את הנתון ומוציאה אותו דרך פורט OUT-1 לנוריות. בצע תוכנית זו בלולאה אינסופית.

מהלך הניסוי

6.6 בנית D/A

מטרת הניסוי היא לבנות D/A בעזרת מגבר שרת + רשת נגדים מסוג סולם של R ו-2R, וליצור אותות שונים על ידי המיקרו-מחשב.

1) בנה את המעגל המתואר בציור 6-2. השתמש בנגדים של 5K ו-10K. את יציאת המעגל חבר לכניסת מגבר מהפך בעל הגבר יחידה.

2) את כניסות המעגל חבר ליציאותיו של PORT OUT-1.

3) קבע את כתובות הפורטים בעזרת מעגל הפענוח.

4) הוצא מספרים בינאריים שונים ל- PORT OUT-1 ומדוד את המתח שביציאת ה-D/A.

5) כתוב תוכנית המוציאה מספרים בינאריים גדלים ל- PORT OUT-1 עם השהייה ביניהם. הרץ את התוכנית בלולאה אינסופית.

6) חבר את יציאת ה-D/A לכניסת משקף התנודות. בדוק את האות המתקבל. צייר את האות בקנה מידה מתאים. בדוק את הליניאריות שלו. חשב גודל מדרגה באות זה.

7) כתוב תוכנית לקבלת גל משולש.

6.7 בנית A/D בעזרת D/A

מטרת הניסוי היא לבנות מעגל A/D בעזרת D/A, כאשר את המספרים הבינאריים הגדלים והבדיקה של המשווה מבצעים בתוכנה.

1) בנה את המעגל המתואר בציור 6-4. ה-D/A הוא המעגל שבנית בניסוי 6.6.

2) כפורט מבוא השתמש ב-PORT IN-1

3) חבר את יציאות PORT OUT-2 ל"לדים" של הערכה.

4) כתוב תוכנית המוציאה מספרים בינאריים גדלים המתחילים מ-0 ל- PORT OUT-1 ובודקת את סיבית D7 ב- PORT IN-1. ברגע שסיבית זו עולה ל-HIGH, מוציאה התוכנית את המספר שב- PORT OUT-1 ל"לדים" (ל- PORT OUT-2).

5) הרץ את התוכנית בלולאה אינסופית.

6) חבר מתחים שונים לכניסת ה-A/D ובדוק את התוצאות.

6.8 דגימת אותות באמצעות A/D ושחזורם

מטרת הניסוי היא לדגום אותות בעזרת A/D ולאגור את הערך הבינארי של הדגימות בזכרון. לאחר מכן לשחזר את האותות על ידי העברת הערכים הבינאריים של הדגימות למעגל D/A.

1) חבר למבוא המעגל שבנית בניסוי 6.7 גל סינוס בתדירות של 0.1HZ כאשר VMAX=5V ו-VMIN=0V.

2) כתוב תוכנית כמו בתת סעיף 4) שבניסוי הקודם, הדוגמת את הכניסה ומוציאה לפלט. יחד עם ההוצאה לפלט, מוציאה התוכנית את המספר לטבלה. בין דגימה לדגימה בצע השהייה בתוכנית של 0.5 שניה.

3) הרץ תוכנית זו למשך 10 שניות והתבונן בפלט.

4) העתק למחברתך את המספרים שקבלת בטבלה. המר אותם לעשרוני. שרטט גרף על ניר מילימטרי.

5) כתוב תוכנית המוציאה את המספרים האגורים בטבלה, בזה אחר זה, ל-D/A מבלי להתייחס לסיבית D7 שב-PORT IN-1. בין הוצאת כל מספר מתבצעת השהיה בתוכנית של 0.5 שניה.

6) חבר את כניסת המשקף למוצא ה-D/A.

7) הרץ את התוכנית והתבונן בצורה המתקבלת במסך. שרטט צורה זו.

8) שנה את ההשהיה בין הוצאת המספרים להשהיה של 0.05 שניה. חזור על סעיף 7). לאיזה מסקנה אתה מגיע מתוצאות ניסוי זה.

6.9 שימוש ב-ADC0808

מטרת הניסוי היא להשתמש ב-A/D מונוליתי בעל כניסות מרובבות - ה-ADC0808.

1) חבר את ה-ADC0808 ל-BUS המערכת בהתאם לשאלת הכנה מס' 5). שים לב שאינך יכול להשתמש במעגל הפענוח של הערכה. חבר מתחי ייחוס (VREF) מתאימים (ראה דפי נתונים).

2) חבר את יציאות PORT OUT-1 ל"לדים" של הערכה.

3) חבר מתח נמדד לערוץ 3 של הרכיב.

4) הרץ את התוכנית שכתבת בשאלת הכנה מס' 6.

5) השווה בין התוצאה הבינארית שהתקבלה ב"לדים" למדידת מתח שתבצע בעזרת מד מתח. שנה את מתח הייחוס עד לקבלת מדידה זהה.

6) שנה את המתח הנמדד והשווה שוב את התוצאות מבלי לשנות הפעם את מתח הייחוס.

פרק 7 - תקשורת בין מחשבים

7.1 נושאי הפרק

א) הכרת הסיווג של שיטות התקשורת. סנכרונית ואסינכרונית, מקבילית וטורית, על ידי FOLLING ועל ידי בקשת פסיקה.
ב) ביצוע תקשורת מקבילית אסינכרונית באמצעות פורטים.
ג) ביצוע תקשורת מקבילית על ידי פסיקה.
ד) ביצוע תקשורת טורית אסינכרונית.
ה) בנית פרוטוקול תקשורת.

רקע עיוני

7.2 סיווג שיטות התקשורת

בתקשורת בין מחשבים, קשורים המחשבים ביניהם דרך קוי תקשורת. בכל שלב בתקשורת קיים מחשב משדר ומחשב מקבל. המשדר שולח את האינפורמציה שלו דרך פורט מוצא והמקבל מקבל אינפורמציה זו דרך פורט מבוא. יתכן מצב שהמחשב המשדר ישדר ולאחר מכן יעבור למצב קליטה ולהיפך, אבל, לא יתכן שיהיה בשני המצבים בו זמנית. כל מחשב אינו רואה את הנעשה אצל משנהו. הם מבחינים רק באינפורמציה הנמצאת בפורטי הכניסה שלהם, לכן חלק מהאינפורמציה המועברת הוא הודעות הקשורות למצבי המחשבים המשדרים והמקבלים, הודעות כמו "מוכן לקבלת הודעה", "קבל הודעה", "סוף הודעה" וכו'.

שיטות התקשורת למיניהם מסווגות בשלש צורות מיון:

א. סינכרונית ואסינכרונית

בתקשורת סינכרונית, קשורים המחשבים לקו משותף המספק אותות סנכרון לשניהם. אותות הסנכרון מאפשרים למחשבים לדעת מתי לשדר ומתי לצפות להודעה בקוי התקשורת. כל מחשב, לפני שהוא משדר הודעה, הוא ממתין להיווצרות אות הסנכרון ואז שולח אותה. מחשב, שאמור לקבל את ההודעה, ממתין להופעת אות הסנכרון ואז אוסף אליו את הנמצא בקוי פורט המבוא שלו.

בתקשורת אסינכרונית חוסכים את הקו של פולסי הסנכרון ומחולל הפולסים. על קוי האינפורמציה שולחים בתחילת כל הודעה אות התחלה. המחשב המקבל ממתין לקבלת אות התחלה זה. כאשר הוא מוצא אות זה הוא אוסף את ההודעה בהמשך. תקשורת זו מקובלת יותר.

ב. מקבילית וטורית

בתקשורת מקבילית שולחים את המידע במקביל. נתון בן 8 סיביות נשלח דרך כבל בן 8 גדים כאשר כל סיבית נשלחת על גיד נפרד בו זמנית. שיטה זו מחייבת כבל בעל מספר גידים רב.

בתקשורת הטורית משתמשים במספר גידים קטן. את הנתון שולחים דרך קו אחד, סיבית אחר סיבית. המשדר והמקבל חייבים להיות מתואמים מבחינת קצב השידור והקליטה.

ג. POLLING או פסיקה

הבעיה בתקשורת היא לדעת מתי מתחיל הדו-שיח. אחת השיטות היא לקבוע את אחד המחשבים כשולט (MASTER) והאחרים כשלוחות (SLAVES). השולט תמיד יוזם את התקשורת. הוא פונה אל השלוחה ושואל אם יש לה מה למסור, וממתין זמן מסוים לקבלת הודעה מהשלוחה. אם לא קיבל, חוזר לתוכנית הראשית שלו. הפניה שלו אל השלוחה נעשית כל מירווח זמן מוגדר. כאשר לשלוחה יש מה למסור, היא ממתינה לפניה של השולט אליה, ושולחת לו חזרה הודעת פתיחה. השולט מגיב חזרה ומתנהל הדו-שיח. שיטה זו נקראת שיטת תקשורת על ידי POLLING.

שיטה אחרת להתחלת שיחה היא בעזרת פסיקה. משתמשים בפורטי מבוא בעלי קו STB (STROBE). כאשר מחשב אחד רוצה לפנות למחשב אחר, הוא שולח לפורט הכניסה שלו את ההודעה יחד עם דופק STROBE. פורט המבוא אוסף אליו את הנתון ומבצע פסיקה למחשב המקבל. האחרון עובר לביצוע תוכנית הפסיקה, המטפלת בנתון הנמסר. שיטה זו מהירה ונוחה אם כי מחייבת שימוש בפורטים מתאימים ובתוכניות פסיקה.

מושג נוסף שקיים בתקשורת הוא המושג HANDSHAKE - לחיצת יד. במושג זה מתכוונים ששולח הודעה ממתין לאישור ממקבל ההודעה על קבלתה. ללא אישור זה אין הוא ממשיך בתוכנית.

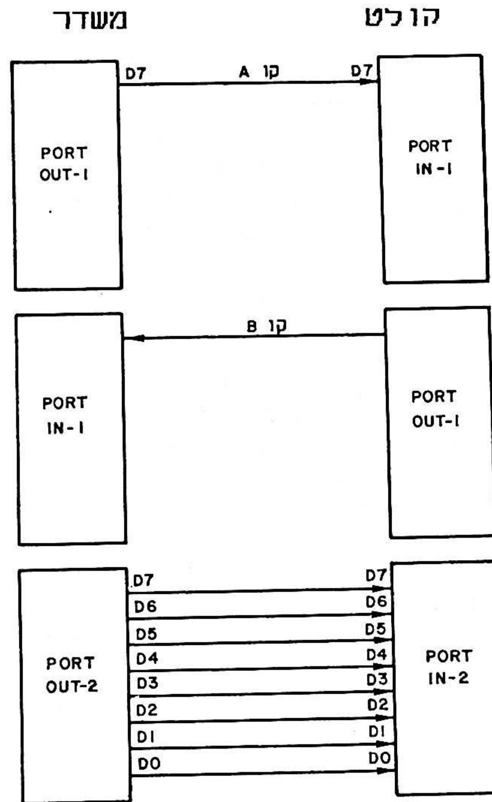
נתאר חלק מהשיטות בעזרת דוגמאות שיבנו במהלך הניסוי.

7.3 תקשורת מקבילית אסינכרונית

נתאר תקשורת בין שתי מערכות מחשבים. האחת יוזמת התקשורת ומשדרת והאחרת מקבלת. זו שמשדרת, משדרת נתונים מטבלה בזה אחר זה, כאשר לסמון נתון אחרון משמש הנתון OD (CARRAGE RETURN - CR). המערכת המקבלת, קולטת את הנתונים ומעבירה אותם לטבלה. ברגע שנמסר הנתון האחרון, מסיימות שתי המערכות.

הגישור בין המערכות יראה בצורה הבאה

קו A משמש את המערכת המשדרת להודיע על מסירת נתון.
קו B משמש את המערכת המקבלת להודיע על קבלת הנתון ועל נכונותה לקבל נתון חדש.



ציור 1 - 7 : תיאור מערכת תקשורת בין שני מחשבים.

נוהל תקשורת 1

מקבל

1) קובע IX (H,L) לכתובת ההתחלה של הטבלה לקליטת הנתונים.
2) מוריד קו D7 ב-OUT-1 ל-LOW (מוכן לקבלת נתון).
3) ממתין לעלית קו D7 ב-PORT IN-1 ל-HIGH. (משדר מסר נתון). מבצע לולאת בדיקה.
4) אוסף נתון מ-PORT IN-2.
5) מעביר אותו לטבלה.
6) מעלה קו D7 ב-PORT OUT-1 ל-HIGH (אישור קבלת נתון).
7) בודק ירידת D7 ב-PORT IN-1 ל-LOW (סיום שדור נתון). לולאת בדיקה.
8) בודק האם הנתון האחרון שהתקבל היה OD. אם כן דלג ל - 11).
9) מגדיל IX (H,L) ב-1.
10) חזור ל - 2).
11) סיים.

משדר

1) קובע IX (H,L) לכתובת ההתחלה של טבלת הנתונים לשדור.
2) ממתין לירידת קו D7 בפורט IN-1 ל-LOW (מקבל מוכן לקבלת נתון). מבצע לולאת בדיקה.
3) מוציא נתון ל-PORT OUT-2.
4) מעלה סיבית D7 ב-PORT OUT-1 ל-HIGH (נמסר נתון).
5) ממתין לעלית קו D7 ב-PORT IN-1 ל- HIGH (אישור קבלת נתון). מבצע לולאת בדיקה.
6) מוריד קו D7 ל-LOW (סיום שדור נתון).
7) בודק האם הנתון האחרון ששודר היה OD (C.R.). אם כן דלג ל- 10).
8) הגדל IX (H;L) ב-1.
9) חזור ל - 2).
10) סיים.

יש לזכור שנוהל התקשורת נקבע שרירותית על ידי מתכנו המערכת. ניתן להוסיף למערכת שתוארה בציור 7-1 את PORT IN-2 למשדר ו-PORT OUT-1-1 למקבל.את שני הפורטים לחבר ביניהם ב-8 חוטים. להוסיף לנוהל התקשורת כך, שכאשר מערכת אחת מסיימת להעביר טבלת נתונים לשניה הן מתחלפות בתפקידים.

7.4 תקשורת מקבילית על ידי פסיקה

השיטה הקודמת מחייבת את המחשב המקבל לטפל רק בפניות של המחשב המשדר. עד לסיום כל ההודעה הוא ממתין לקבלת נתונים ואינו יכול לעשות כל דבר אחר. המחשב המשדר, כיוזם התקשורת, יכול לעסוק גם בדברים אחרים.

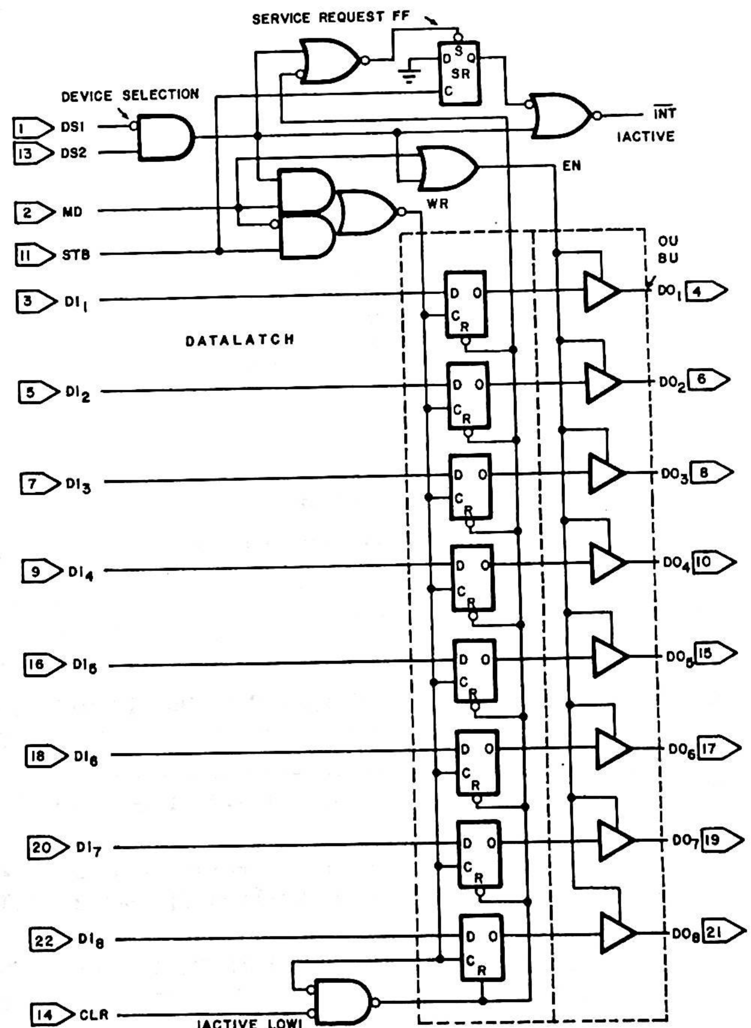
תקשורת על ידי פסיקה משחררת את המחשב המקבל לטפל בתוכנית אחרת מלבד תוכנית התקשורת, אבל היא מחייבת שימוש בפורט היוצר בקשת פסיקה. פורט שמסוגל לעשות זאת הוא הרכיב 8212, המתואר בציור 7-2.

CLR	$(\overline{DS_1} DS_2)$	STB	•SR	INT
0	0	0	1	1
0	1	0	1	0
1	1	⌐_	0	0
1	1	0	1	0
1	0	0	1	1
1	1	⌐_	1	0

•INTERNAL SR FLIP-FLOP

STB	MD	$(\overline{DS_1} DS_2)$	DATA OUT EQUALS
0	0	0	3-STATE
1	0	0	3-STATE
0	1	0	DATA LATCH
1	1	0	DATA LATCH
0	0	1	DATA LATCH
1	0	1	DATA IN
0	1	1	DATA IN
1	1	1	DATA IN

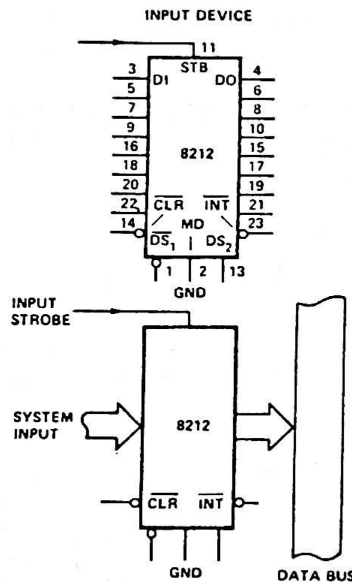
CLR RESETS DATA LATCH
SETS SR FLIP-FLOP
NO EFFECT OUTPUT BUFFER

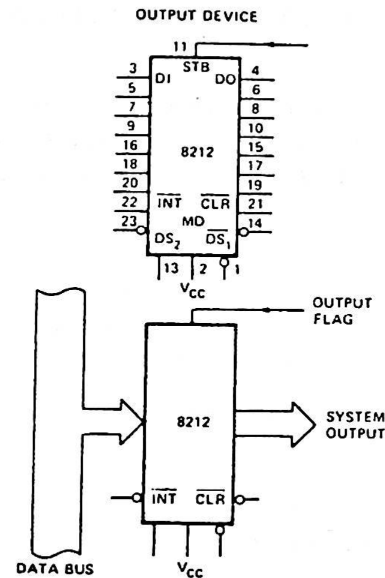


ציור 2 - 7 : תיאור ה - 8212.

ה - 8212 כולל תאי LATCH שיציאותיהם נשלטות על ידי חוצצים. לרכיב 2 קוי CS (DS1 ו-DS2). מלבד מרכיבים אלה המאפיינים פורטים, יש לו קוים נוספים. קו CLR הגורם לאיפוס תוכו תאי ה-LATCH. קו MD הקובע אם הוא עובד ב-MODE OUT (MD=VCC) או ב-NODE IN (MD=O).

הקוים המאפשרים ביצוע פסיקה הם הקוים STB ו-INT. את קו INT מחברים לקו פסיקה של ה-CPU. אם קו הפסיקה מופעל ב-HIGH, מחברים אותו דרך מהפך. הורדת קו STB מ-HIGH ל-LOW גורמת לבקשת פסיקה בקו INT.





ציור 7-3 :

חיבור ה-8212 כפורט מבוא

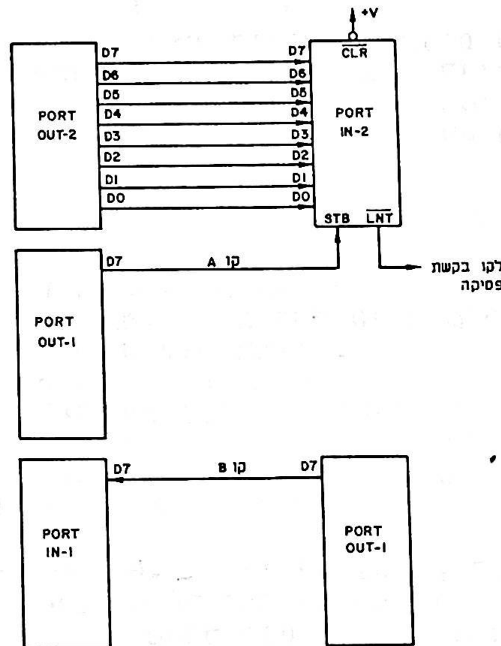
ציור 7-4 :

חיבור ה-8212 כפורט מוצא

בציור 7-3 מתואר חיבור ה-8212 כפורט מבוא. במקרה זה, ירידת קו STB ל-LOW גורמת לנעילת האינפורמציה שבקוי הכניסה לרכיב (INPUT SYSTEM) בתאי ה-LATCH ולבקשת פסיקה. את האינפורמציה מספקת מערכת חיצונית. ה-CPU בתוכנית הפסיקה פונה לפורט, אוסף את הנתון , מעבד אותו במידת הצורך ויכול לחזור ולהמשיך בתוכנית הראשית.

בציור 7-4 מתואר חיבור ה-8212 כפורט מוצא. ירידת קו STB יוצרת בקשת פסיקה, אבל לא מכניסה דבר לתאי ה-LATCH (MD=1, ראה ציור 7-2). בתוכנית הפסיקה, מוסר ה-CPU נתון לפורט. אז יוצר ה-STB אוסף אליו את הנתון. קו STB מחובר בכל מקרה למערכת החיצונית ל-CPU גם בשמוש בפורט מוצא וגם בפורט מבוא.

צורת החיבור לתקשורת מהסוג הזה תיראה כמו בציור 7-5.



ציור 5 – 7 : מעגל לתקשורת באמצעות פסיקה.

קו A משמש את המשדר לביצוע STB בפורט המבוא של המקבל. STB זה גורם לבקשת פסיקה אצל המקבל.
קו B משמש את המקבל להודיע על אישור קבלת הנתון ונכונותו לקבל נתון חדש.

נוהל התקשורת יהיה כדלקמן.

נוהל תקשורת 2

מקבל

1) קובע IX (H,L) לכתובת ההתחלה של הטבלה לקליטת נתונים.
2) מאחסן את IX (H,L) במקום מוגדר.
3) הורדת דגל הפסיקה.
4) תוכנית ראשית. ריצה בלולאה אינסופית.

תוכנית פסיקה

5) העלאת דגל הפסיקה.
6) העלאת קו D7 (תפוס) ב-PORT OUT-1 ל-HIGH.
7) אוסף נתון מ-PORT IN-2.
8) הוצאת IX (H,L) ממקום האיחסון.
9) הכנסת הנתון לטבלה.
10) העלאת IX (H,L) ב-1.
11) אחסון IX (H,L) במקום המוגדר.
12) הורדת קו D7 ב-PORT OUT-1 ל-LOW.
13) הורדת דגל פסיקה.
14) חזרה לתוכנית הראשית.

משדר

1) קובע IX (H,L) לכתובת ההתחלה של הטבלה לקליטת נתונים.
2) מוסר נתון דרך PORT OUT-2.
3) מוסר STB דרך PORT OUT-1.
4) ממתין לעלית קו D7 ב-PORT IN-1 ל-HIGH (אישור קבלת נתון). מבצע לולאת בדיקה.
5) ממתין לירידת קו D7 ב-PORT IN-1 ל-LOW. (מוכן לנתון נוסף). מבצע לולאת בדיקה.
6) בדיקה האם הנתון האחרון ששודר היה OD (C.R.). אם כן דלג ל- 9).
7) הגדל IX (H,L) ב-1.
8) חזור ל- 2).
9) עצור.

רצוי שהתוכנית הראשית במקבל תהיה תוכנית המציגה ערכים משתנים בפלט.
לדוגמא, חיבור PORT OUT-2 במקבל לנוריות והתוכנית הראשית תבצע ספירה בינארית בנוריות הפלט עם השהיה. הרצת תוכנית המשדר תראה שאין הפרעה כמעט בתוכנית הראשית. לאחר שהמשדר מסיים את השידור ניתן לעצור את התוכנית הראשית של המקבל ולבדוק את הנתונים שנמסרו לו.

7.5 תקשורת טורית אסינכרונית

תקשורת טורית היא תקשורת בה מתרגמים את הנתון בן 8 הסיביות לסדרה של דפקים טוריים, אפס ואחד, המשודרים דרך קו תקשורת אחד. המקבל יודע את משך הזמן של כל דופק שמשדר המשדר.

בתקשורת טורית אסינכרונית קיימת בעיה של זהוי ההתחלה של כל נתון. לכן קבעו את הנוהל הבא:

סיבית התחלה - START BIT

הקו במצבו הנורמלי נמצא ב-HIGH. לפני כל נתון, הנשלח בצורה טורית, חייבת לבוא סיבית אפס למשך זמן המוקצב להעברת כל סיבית. סיבית זו נקראת סיבית התחלה. מקבל הנתונים מזהה התחלת שדור תו על ידי זהוי המעבר מ-"1" ל-"0".

סיביות הנתון - DATA BITS

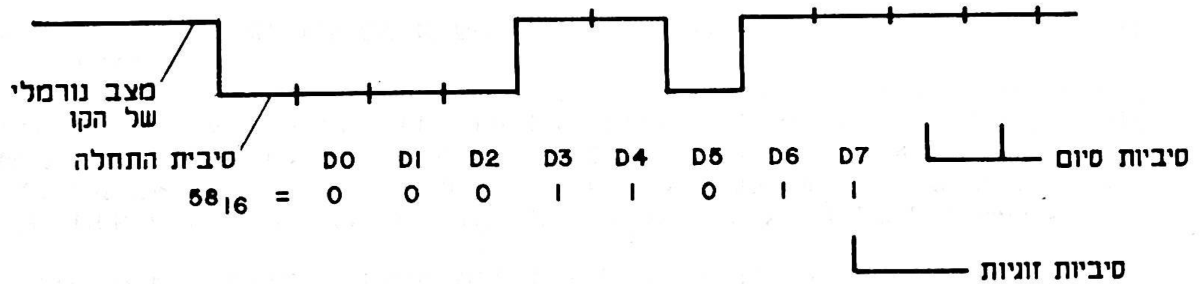
עם סיום השדור של סיבית ההתחלה, משודרות 7 סיביות הנתון בזו אחר זו. כל אחת במשך זמן מוגדר וקבוע. מכיוון שמקבל הנתונים יודע מתי התחיל השדור, הוא מבצע את דגימת סיביות הנתון באמצע של כל סיבית. זאת כדי להתגבר על הבעיה של תופעות מעבר.

סיבית בקורת - PARITY BIT

בדרך כלל, ממירים כל נתון לצופן ASCII ואותו משדרים. צופן ASCII הוא צופן בעל שבע סיביות. בסיבית השמינית משתמשים לבדיקת זוגיות. סיבית זו משמשת את המקבל לבדיקת הנכונות של הנתון שקיבל. ערך הסיבית ("0" או "1") נקבע לפי מספר ה-"1" שבנתון. קיימות שתי שיטות. ביקורת זוגיות וביקורת פרט. בביקורת זוגיות, מספר ה-"1", כולל סיבית הבקורת חייב להיות זוגי. לדוגמא, אם בנתון יש שלשה "1", קובע המשדר את סיבית הביקורת ל-"1". אם יש ארבעה "1", תהיה סיבית הביקורת "0".
בביקורת פרט, מספר ה-"1", כולל סיבית הביקורת, יהיה פרט.

סיביות סיום - STOP BITS

בסוף כל נתון, משודרות שתי סיביות של "1" לוגי. סיביות אלה משמשות להעברת הקו למצבו הנורמלי, למשך זמן המאפשר למקבל לעבד עבוד ראשוני את הנתון שאסף ולהסתנכרן מחדש על התחלת שדור התו הבא.
לסכום, שדור תו בודד (58H) נראה בצורה הבאה:



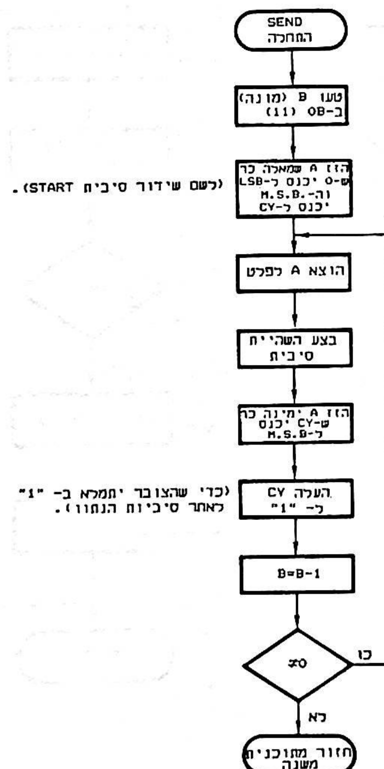
ציור 6 -7 : שדור התו 5816 בתקשורת טורית אסינכרונית.

קצב שדור נמדד ביחידות של BAUD שהם סיביות משודרות לשניה. יחידת קצב שידור אחרת היא סיביות השניה שבה הכוונה לסיביות הנתוו המועברות בשניה. לדוגמא אם משדרים בקצב של 10 תוים לשניה מקבלים 110 BAUD. כל תו דורש לשדורו 11 סיביות שדור (כולל סיבית ההתחלה וסיביות הסיום). קצב זה גם שווה ל-80 סיביות לשניה (סיביות נתוו).

נתאר מספר תרשימי זרימה של תוכניות העוסקות בשידור וקליטה בתקשורת טורית.

תרשים זרימה א'

תרשים זה מתאר תוכנית משנה המשדרת תו אחד הנמצא בצובר A. הנתונים יוצאים מסיבית ה-.L.S.B של פורט מוצא.



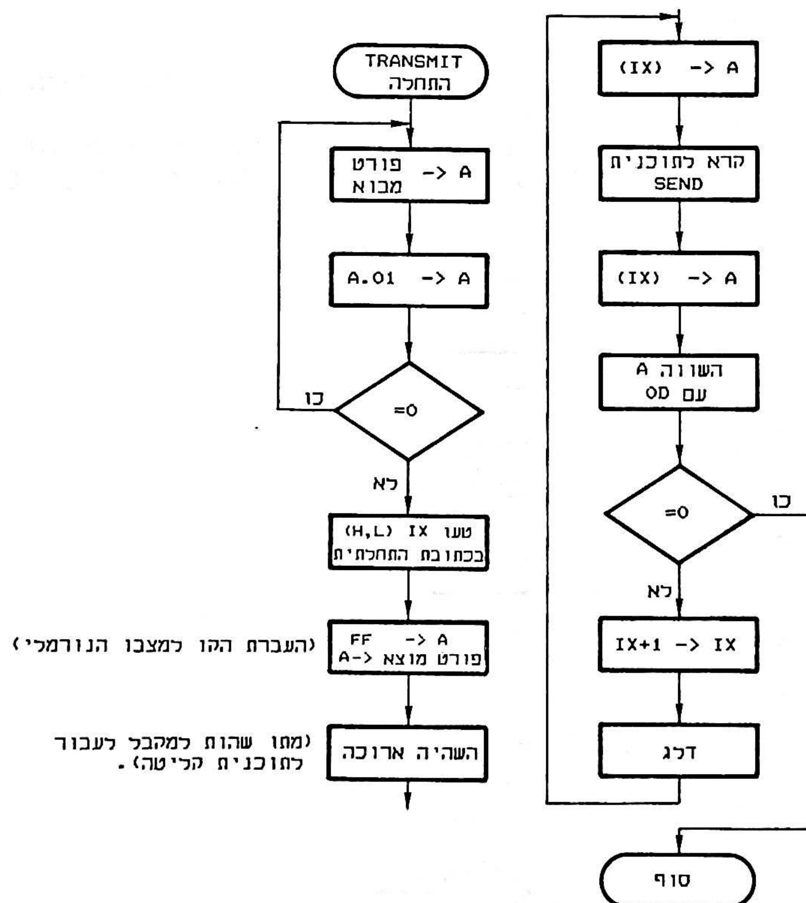
בתוכנית קליטה קיימת בעיה שהיא עלולה להתחיל לקלוט מאמצע של שדור בלוק נתונים.

קיימות מספר שיטות לפתרון הבעיה. האחת, לקבוע נתון מסוים שתמיד ישמש כנתון ראשון. עד שהקולט לא מזהה נתון זה, הוא אינו אוסף אליו את שאר הנתונים. אפשרות שניה היא להשתמש בדגל שיעלה ל "1" כאשר הוא מוכן לקלוט. כל זמן שדגל זה למטה, לא ישדר המשדר.

נשתמש בשיטה השניה. נחבר סיבית D0 בפורט מוצא של המקבל, לסיבית D0 בפורט מבוא של המשדר.

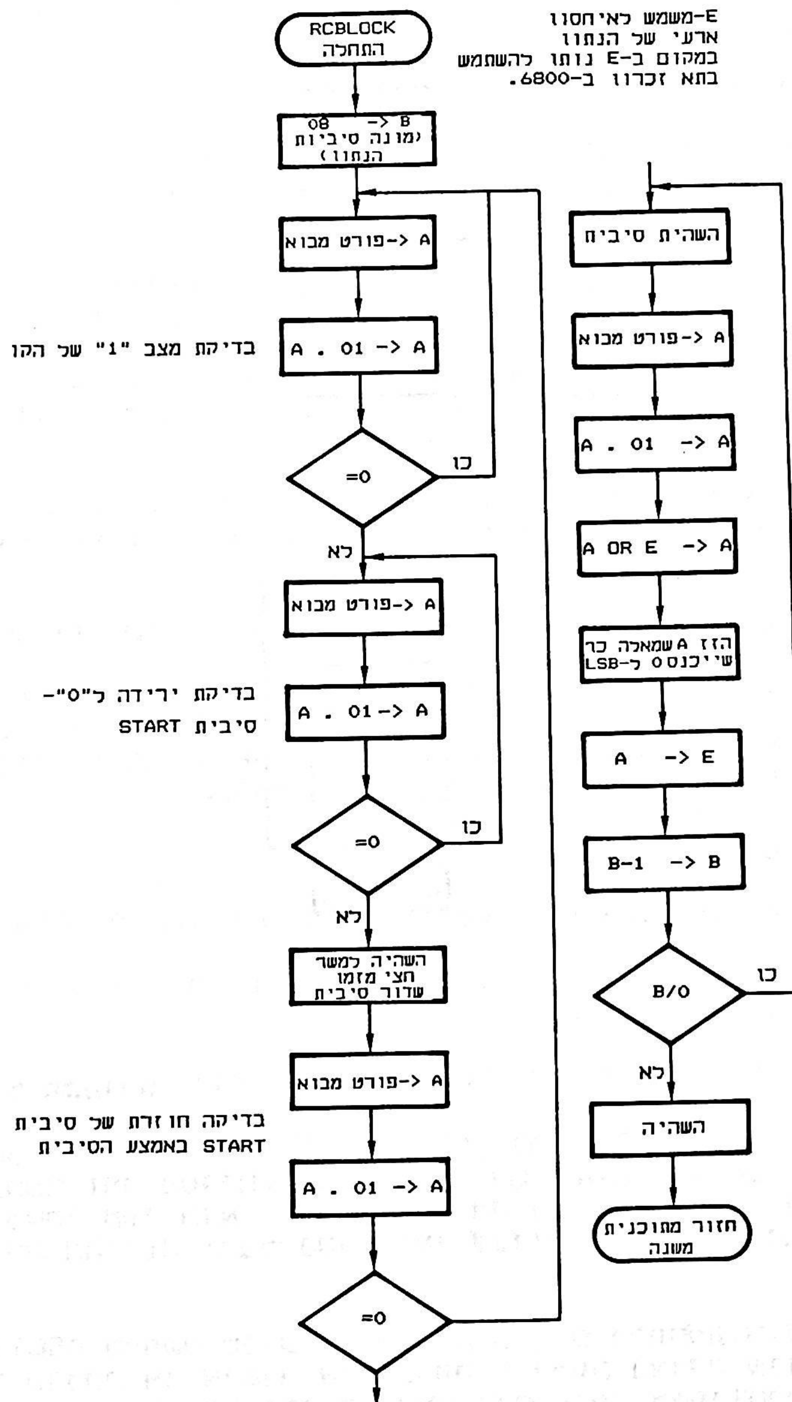
תרשים זרימה ב'

תרשים זה מתאר תוכנית המשדרת בלוק נתונים המסתיימים ב - OD. התוכנית ממתינה לקבלת אות "מוכן" מהמערכת המקבלת. האות מתקבל דרך סיבית D0 בפורט מבוא. לאחר קבלת האות, עוברת התוכנית לשדור בלוק הנתונים.



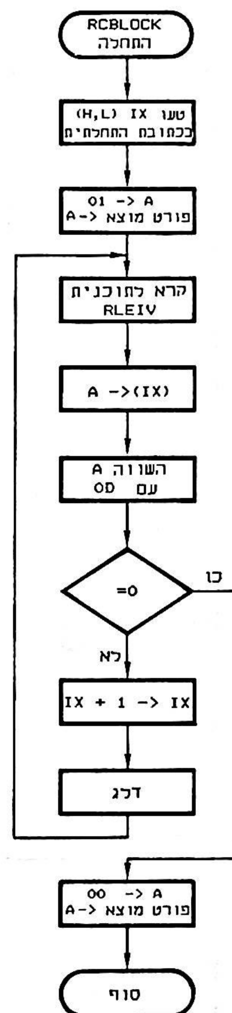
תרשים זרימה ג'

תרשים זה מתאר תוכנית משנה לקליטת תו בתקשורת טורית דרך קו ה-L.S.B של פורט מבוא.



תרשים זרימה ד'

תרשים זה מתאר תוכנית לקליטת בלוק נתונים. התוכנית מוציאה את "מוכן" בתחילתה ומסתיימת עם קבלת OD.



7.6 פרוטוקול תקשורת

כאשר שני מחשבים מנהלים דו-שיח ביניהם, נמצא האחד במצב שדור והאחר במצב קליטה במשך זמן מסוים. לאחר מכן הם מתחלפים. כל מחשב צריך לדעת בצורה כלשהי מתי הוא יכול לשדר, מה לעשות עם הנתונים המגיעים אליו, לזהות אם ההודעה שקבל נכונה או שגויה ואם היא מופנית אליו בכלל.

שדור בצופן ASCII מאפשר שמוש בצפנים מוסכמים להתחלת הודעה, סיום הודעה, אישור קבלתה או אישור אי קבלתה (קבלתה בצורה שגויה) וכד'. שדור מספר בינארי נעשה על ידי כך שמפרקים אותו לשתי ספרות הקסאדצימליות, ומשדרים אותו בצופן ASCII זו אחר זו.

בכל תקשורת יש יוזם התקשורת, שנקבע על ידי בקשת פסיקה שהוא מבקש או מקביעה מראש של אחד המחשבים.

נדגים זאת בפרוטוקול הבא, שיתאר דו שיח בין שני מחשבים. מחשב A נקבע כיוזם התקשורת ומחשב B כשלוחה של A. B אינו יכול לשדר לפני ש-A פונה אליו ומשדר לו הודעת פתיחה.
הצפנים הראשונים של צופן ASCII (0-1F) משמשים כצפני הודעות מקובלות - תוי בקרה.

הודעת הפתיחה של A תכלול בדרך כלל:
1) תו בקרה
2) מספר ש-B ישייך אותו אליו (מספר מנוי למשל).
3) תו בקרה המסמן "קבל הודעה" או "מסור הודעתך".
4) תו סיום הודעת הפתיחה.

לאחר שדור הודעת הפתיחה, עובר A למצב המתנה להודעה מ-B למשך זמן קצוב. B בודק ומפענח כל אחד מהמספרים המקבלים. הוא מזהה שהודעת הפתיחה מכוונת אליו והוא מפענח את סעיף 3). לאחר מכן הוא ממתין זמן מסוים כדי לאפשר ל-A לעבור לקליטה לפני שהוא משדר, ואז משדר אליו את הודעתו. גם הודעתו מתחילה ומסתיימת בתוי בקרה בצופן ASCII.

A מקבל את ההודעה של B, מפענח אותה ומשדר לו הודעת ACKNOWLEDGE או NOT ACKNOWLEDGE (מאשר או לא מאשר) בהתאם לבדיקת נכונות ההודעה.

B חייב לאשר גם הוא את אישורו של A לפני שמסתיים הדו-שיח.

נוהל התקשורת, נקבע על ידי מתכנן המערכת. הוא כותב את תוכניות התקשורת של A ושל B. בנושא זה אין נוהלים וכללים מוסכמים ואיש הישר בעיניו יעשה. במצב היום, אין מערכות של חברות שונות יכולות לדבר זו עם זו, אלא אם יתאימו אחת אל האחרת.

7.7 שאלות הכנה

1) רשום תרשימי זרימה למקבל ולמשדר בהתאם לנוהל תקשורת 1.

2) המר את תרשימי הזרימה של שאלה 1) לתוכניות מחשב, בהתאם לסוג המיקרופרוססור שברשותך.

3) רשום תרשימי זרימה למקבל ולמשדר בהתאם לנוהל תקשורת 2.

4) המר את תרשימי הזרימה של שאלה 3) לתוכניות המחשב, בהתאם לסוג המיקרופרוססור שברשותך.

5) המר את תרשים זרימה א' לתוכנית מחשב בהתאם לסוג המיקרופרוססור שברשותך.

6) המר את תרשים זרימה ב' לתוכנית מחשב.

7) המר את תרשים זרימה ג' לתוכנית מחשב.

8) המר את תרשים זרימה ד' לתוכנית מחשב.

מהלך הניסוי

הניסויים בתקשורת יתקיימו בין שני צוותים על שתי ערכות נסויים, כאשר צוות אחד משדר והצוות האחר קולט.

7.8 תקשורת מקבילית אסינכרונית

מטרת הניסוי היא לקיים תקשורת מקבילית אסינכרונית בין שני הצוותים, בהתאם לנוהל תקשורת 1. צוות אחד משדר בלוק נתונים והצוות האחר קולט ומאחסן בלוק נתונים זה.

1) קבע ערכת נסויים אחת כמשדרת וערכה שניה כקולטת. בצע חיבורים ביניהם כמתואר בציור 7-1.

2) כתוב תוכנית למערכת הקולטת בהתאם לנוהל תקשורת 1.

3) כתוב תוכנית למערכת המשדרת בהתאם לנוהל תקשורת 1.

4) הכנס 2 נתונים לטבלת הנתונים במערכת המשדרת. כנתון אחרון הכנס את המספר 0D.

5) הרץ את התוכניות בצעד יחיד. שיטת ההרצה היא כדלקמן .
א) הרץ את תוכנית המשדר עד לשלב שבו היא ממתינה לתגובה מהמקבל. בדוק שאינה ממשיכה מעבר לכך.
ב) עבור להרצת תוכנית המקבל עד שהיא מגיעה לשלב שבו היא ממתינה לתגובה מהתוכנית המשדרת. בדוק שאינה ממשיכה מעבר לכך.
ג) המשך בתוכנית המשדר וכן הלאה עד לשלב בו שתי התוכניות נעצרות.

6) לאחר בדיקת שתי התוכניות, מלא את טבלת הנתונים שמספר גדול יותר של נתונים. הרץ את שתי התוכניות.

7) בדוק שהנתונים מהמשדר הגיעו לטבלה המתאימה במקבל.

8) הוסף בהמשך לתוכנית המקבל תוכנית שדור מטבלת נתונים אחרת.

9) הוסף בהמשך לתוכנית המשדר תוכנית קליטת נתונים והעברתם לטבלה.

10) חבר את PORT OUT-2 בערכה הקולטת ל- PORT IN-2 במשדר.

11) שרטט את המעגל השלם.

12) הרץ את שתי התוכניות, כך שהמשדר והמקבל מחליפים תפקידים בסיום שלב א'.

13) בדוק את שתי הטבלאות שהתקבלו במקבל ובמשדר.

14) רשום בצורה מסודרת, כולל תיאורים ותרשימי זרימה, את תוכניות השדור והקליטה.

7.9 תקשורת מקבילית באמצעות פסיקה

מטרת הניסוי היא לבצע תקשורת מקבילית בין שני הצוותים בהתאם לנוהל תקשורת 2. ה-CPU המקבל קשור גם בעזרת קו פסיקה לפורט המבוא שלו. CPU זה רץ על תוכנית ראשית. עם קבלת בקשת פסיקה אוסף את הנתון אליו וממשיך בתוכנית הראשית.

1) וודא שקו INT שבמערכת גורם לבקשת פסיקה במערכת המיקרו-מחשב שברשותך. כן וודא שיש באפשרותך לכתוב תוכנית פסיקה, אליה יפנה המחשב כאשר תוריד את קו INT ל-LOW.

2) בנה את המעגל המתואר בציור 7-5. חבר את יציאת INT של IN-2 PORT שבמקבל לקו INT שבמערכת. במידה ובמערכת שברשותך קו INT מגיב ל-HIGH, חבר קו זה דרך מהפך.

3) חבר את PORT OUT-2 של המקבל לנוריות.

4) רשום תוכנית ראשית למקבל בהתאם לנוהל תקשורת 2. תוכנית זו תבצע בסעיף 4) שלב הוצאת מספרים בינאריים גדלים בלולאה אינסופית לנוריות. בין הפניות לפורט המוצא תבצע השהיה.

5) רשום תוכנית פסיקה למקבל כמתואר בנוהל תקשורת 2.

6) רשום תוכנית שדור למשדר.

7) הרץ את תוכנית מקבל. הנוריות צריכות להראות ספירה בינארית אינסופית (לאחר FF ממשיכה ב-00).

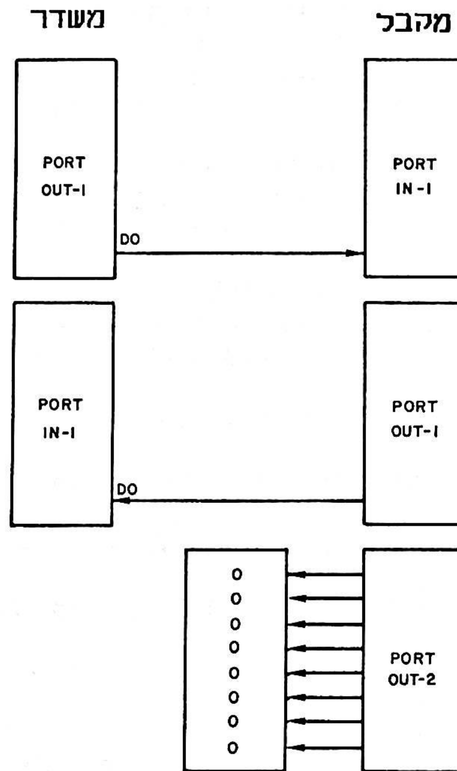
8) הרץ את תוכנית המשדר. הרצת תוכנית המשדר עד לסיומה לא תראה כל השפעה על הספירה הבינארית.

9) לאחר סיום תוכנית המשדר, עצור את המקבל ובדוק אם הנתונים התקבלו לטבלה.

7.10 תקשורת טורית אסינכרונית

מטרת הניסוי לבצע תקשורת טורית אסנכרונית בין שתי ערכות נסויים בעזרת פורטים מקביליים. המערכת המקבלת מודיעה על "נכונותה" בעזרת קו מיוחד לשם כך. את הנתונים המתקבלים מוציאה המערכת המקבלת לנוריות.

1) בנה את המעגל הבא



ציור 7-7 : מעגל לתקשורת טורית באמצעות פורטים מקביליים.

2) רשום במערכת המשדרת תוכנית המבצעת את תרשים זרימה ב'. תוכנית זו נעזרת בתוכנית המשנה המתוארת בתרשים זרימה א'. קבע את ההשהיה שבתרשים זרימה א' ל-1 שני לפחות.

3) רשום במערכת המקבלת תוכנית המבצעת את תרשים זרימה ד'. תוכנית זו נעזרת בתוכנית המשנה המתוארת בתרשים זרימה ג'. קבע את ההשהיה בתוכנית זו לאותה השהיה של תוכנית השדור. הוסף לתרשים זרימה ג' הוצאה לנוריות הפלט של תוכן A לפני העברה לאוגר E. בצורה זו תוכל לראות בצורה איטית את ההעברה של הנתונים.

4) קבע טבלת נתונים להעברה במשדר. נתון אחרון יהיה 0D.

5) הרץ את תוכנית המשדר.

6) הרץ את תוכנית המקבל ובדוק זרימת הנתונים בצורה טורית אל המקבל.

7) בסיום ההעברה, בדוק את בלוק הנתונים במקבל.

פרק 8 - מדידת זמן ושעון אמיתי

8.1 נושאי הפרק

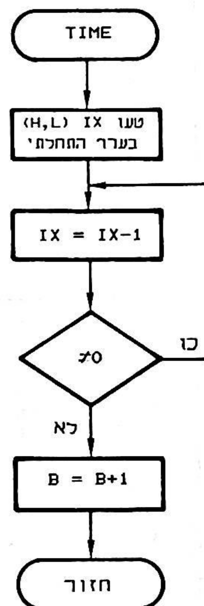
א. הכרת מדידה של זמן בין שני מאורעות בתוכנה.
ב. הכרת מדידה של זמן בין שני מאורעות באמצעות פסיקה.
ג. הכרת שעון זמן אמיתי ואופן בניתו.

רקע עיוני

8.2 מדידת זמן בין שני מאורעות בתוכנה

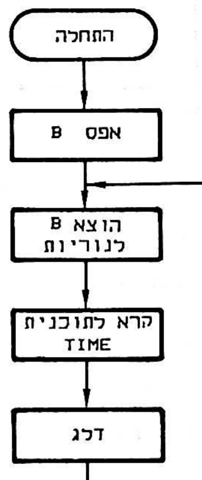
אחת השיטות לביצוע מדידת זמן היא על ידי הרצת ה - CPU בלולאות אינסופיות. מדידה זאת נעשית בתוכנה בלבד. לולאה אחת היא לולאה בעלת אורך זמן קבוע, שבסיומה מועלה מונה ב-1. נכנה לולאה זו בשם TIME והיא נראית כך (B משמש כמונה).

תרשים זרימה א'



כדי לראות את המונה ניתן להשתמש בתוכנית הבאה

תרשים זרימה ב'

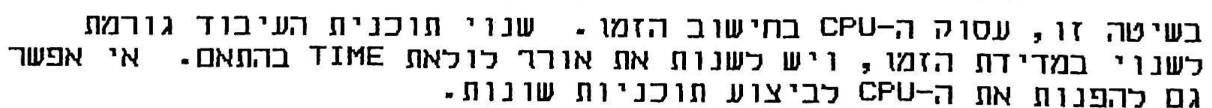


נראה מדידת זמן בין שני מאורעות בעזרת תוכנית TIME זו. נמדוד זמן מחזור של מתנד גל ריבועי. מחברים את יציאת המתנד לכניסת פורט מבוא. נניח לקו D0.
התוכנית תיראה לפי תרשים זרימה ג'.

תרשים זרימה ג'

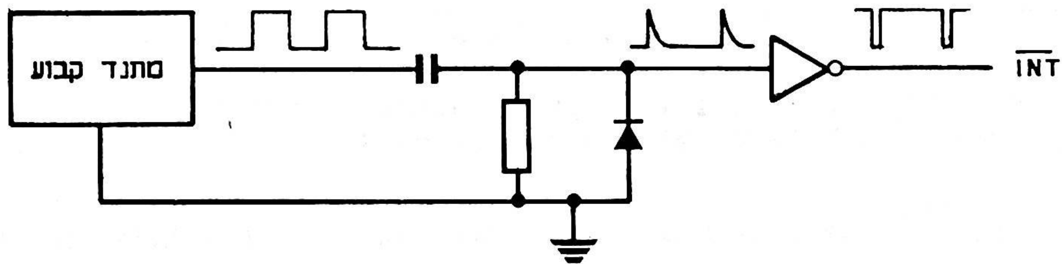
התחלה
אפס B
קרא ל-TIME
קלוט ממתנד ל-A
A = A . 01
≠0
כן
בדיקת מצב "1" לוגי
לא
קרא ל-TIME
קלוט ממתנד ל-A
A = A . 01
≠0
כן
בדיקת מצב "0" לוגי
לא
העבר B לנוריות

בתוכנית זו, לאחר כל לולאה פנימית, דוגם ה-CPU את יציאת המתנד. ברגע שהוא מזהה עליה של האות, לאחר שהיה ב-LOW, הוא מעביר את תוכן המונה ליציאה, מאפס אותו ומתחיל במדידת זמן מחדש. התנאי בתוכנית שהתוצאה ב-B קטנה מ-255.


בשיטה זו, עסוק ה-CPU בחישוב הזמן. שנוי תוכנית העיבוד גורמת לשנוי במדידת הזמן, ויש לשנות את אורך לולאת TIME בהתאם. אי אפשר גם להפנות את ה-CPU לביצוע תוכניות שונות.

8.3 מדידת זמן באמצעות פסיקה

כדי להתגבר על הבעיה שתוארה בסעיף הקודם, משתמשים בשיטה הבאה. מוסיפים למערכת מתנד חיצוני. מתנד זה נותן דופק כל פרק זמן ידוע מראש. את יציאתו מחברים למעגל גוזר, על מנת לקבל פולסים צרים יותר המסמנים את תופעת המעבר בלבד. את יציאת הגוזר, דרך קוטם שיאים שליליים מחברים לקו הפסיקה של ה-CPU. כמתואר בציור (הדיודה קוטמת את השיאים השליליים).



ציור 1 - 10: מעגל שעון (CLOCK).

מעגל המתנד מוריד את הצורך בביצוע הלולאה הפנימית. תוכנית הפסיקה מבצעת את המניה ועבוד התוצאה (כמו במקרה של מניה מעבר לתא בודד).

בצורת עבודה זו משוחרר ה-CPU לביצוע תוכנית ראשית כמו דגימת המאורעות או כל דבר אחר. התוכנית למדידת זמן המחזור של המתנד המשתנה מהסעיף הקודם תיראה כמו בתרשים זרימה ג) רק ללא הפניה לתוכנית משנה TIME. ניתן להוסיף בדיקות נוספות ובכל זאת מדידת הזמן תהיה מדויקת. למשל, לשם חישוב התדירות, יש לחשב F = 1/T. כלומר, להוסיף תוכנית חילוק מתאימה. בשיטה זו אין בעיה להוסיף זאת.

8.4 שעון זמן אמיתי

שעון זמן אמיתי הוא למעשה המערכת שתוארה בסעיף הקודם. תוכנית הפסיקה מבצעת, בדרך כלל, מניה ביותר מתא אחד. שעון העובד על בסיס השעון הרגיל, צריך להתחשב גם בחלוקה של ששים שניות בדקה, ששים דקות בשעה ועשרים וארבע שעות ביממה. למערכת כזו מקציבים מספר תאים לאחסון הערכים השונים. תוכנית הפסיקה, דואגת להוסיף כל פסיקה לתאים את הערך הדרוש. תוכנית כזו מראה את הצורך בפסיקה חיצונית. הזמן הדרוש לחישוב שונה מחישוב לחישוב. זמן החישוב, כאשר יש נשא מספרה לספרה, ארוך יותר מכאשר התוכנית מעלה רק את הספרה הפחות משמעותית.

8.5 שאלות הכנה

1) המר את תרשימי הזרימה א׳ ב׳ ג׳ לתוכניות מחשב בהתאם לסוג המיקרופרוססור שברשותך, כאשר פורט מוצא ופורט מבוא נמצאים בכתובת IO-04.
2) כתוב תוכנית פסיקה לסעיף 10.3, המונה ומאחסנת מספרים בני 16 סיביות.
3) כתוב תוכנית פסיקה לסעיף 10.3, המונה עד 99 בצופן B.C.D בשני תאי זכרון.

מהלך הניסוי

8.6 מדידת זמן מחזור של מתנד משתנה בתוכנה

מטרת הניסוי היא לדגום יציאה של מתנד, ולמדוד את זמן המחזור שלו. ה-CPU רץ על תוכנית שגם דוגמת וגם מבצעת לולאות השהיה למדידת הזמן.

1) חבר את יציאת המתנד המשתנה לכניסת D0 של PORT IN-1.

2) חבר את יציאות PORT OUT-1 ל"לדים" של הערכה.

3) רשום תוכנית בהתאם לתרשים זרימה ג׳.

4) הרץ את התוכנית.

5) חבר משקף תנודות ליציאת המתנד.

6) שנה את ההשהיה בתוכנית TIME כך שתקבל קריאה מתאימה לזמן המחזור.

7) שנה בעזרת הפוטנציומטר את תדירות המתנד. בדוק אם הקריאה החדשה מתאימה למדידה במשקף.

8.7 מדידת זמן באמצעות פסיקה

מטרת הניסוי היא לדגום יציאה של מתנד ולמדוד את זמן המחזור שלו. בניסוי זה עסוק ה-CPU בדגימת מוצא המתנד והוצאת התוצאה לפלט בלבד. מדידת הזמן נעשית בעזרת פסיקות המתקבלות ממתנד קבוע.

1) חבר את יציאת המתנד הקבוע למעגל גזירה וקטימה כמתואר בציור 8-1.

2) חבר את יציאת המעגל דרך מהפך לקו INT בערכה (אם מערכת המיקרו-מחשב שלך קו INT מופעל ב-HIGH ותר על המהפך).

3) רשום תוכנית פסיקה המבצעת העלאת B ב-1 וחזרה.

4) רשום תוכנית ראשית בהתאם לתרשים זרימה ג׳, ללא קריאה לתוכנית TIME.

5) הרץ את התוכנית הראשית.

6) חזור על סעיפים 5) עד 7) שבניסוי הקודם.

8.8 שעון זמן אמיתי

מטרת הניסוי היא ליצור מערכת מניית זמן המתבצעת בעזרת תוכנית פסיקה הסופרת דפקים בתדירות 1HZ, המגיעים ממתנד קבוע. תוכנית הפסיקה סופרת ואוגרת שניות, עשרות שניות, דקות ועשרות דקות.

1) חבר את יציאת המתנד הקבוע למעגל גזירה וקטימה כמתואר בציור 8-1.

2) חבר את יציאת המעגל דרך מהפך לקו INT בערכה (או ללא מהפך במקרה שקו INT מופעל ב-HIGH).

3) רשום תוכנית פסיקה הסופרת ואוגרת בארבעה תאים שניות, עשרות שניות, דקות ועשרות דקות.

4) חבר את יציאות PORT OUT-1 ו- PORT OUT-2 לתצוגת ה- 7-SEG., כמו בניסוי המתואר בפרק 4.

5) רשום תוכנית ראשית הממירה את תוכן ארבעת התאים לצופן 7-SEG. ומציגה אותם בתצוגה בצורה דינמית, בלולאה אינסופית. התוכנית מתחילה באיפוס התאים או בקביעת ערך התחלתי כלשהו להם.

6) הרץ את התוכנית הראשית. בתצוגה עליך לקבל שעון.

